Ref. Ares(2020)5034826 - 25/09/2020

AQUA EXCEL 2020

AQUAculture infrastructures for EXCELlence in European fish research towards 2020 — AQUAEXCEL2020

Virtual laboratory version 2

Finn Olav Bjørnson, Eleni Kelasidi

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 652831. This publication reflects only the view of the author, and the European Commission cannot be held responsible for any use which may be made of the information contained therein

Executive Summary

Objectives:
This deliverable describes the second version of the virtual laboratory developed in WP5 of AQUAEXCEL[2020]. The document describes the technical framework used to integrate the different submodels that are part of the virtual laboratory, short descriptions of these submodels, and demonstrations of the integration through simple simulation experiments.

Rationale: One of the main research activities in AQUAEXCEL[2020] WP5 is to develop a virtual laboratory system that enables virtual experiments in aquaculture research facilities. This system will feature a framework that allows the integration of mathematical models of different subsystems in common simulations, replicating the system operation of research laboratories.

Main Results:
The second version of the virtual laboratory is now complete. We have demonstrated that it is possible to run virtual experiments by using the submodels developed in WP5 in integrated simulations. A web-interface for setting up and executing virtual experiments has been completed and is available for the public at https://ae2020virtuallab.sintef.no/. The interface is linked with the technical framework integrating the models such that users can execute virtual experiments without needing deep insight into how the framework and submodel integration is implemented. For the current version we have made constraints in the web-interface so the user is restricted to co-simulate models that have been pre-selected, however advanced users with direct access to the underlying framework have no restrictions on how they want to combine the submodels.

Authors/Teams involved:
SINTEF: Finn Olav Bjørnson (lead author), Eleni Kelasidi
HCMR team has contributed with the growth model
NTNU/JU team has contributed with the hydrodynamic model
WR/WU team has contributed with the watertreatment model

Table of contents

1. BACKGROUND

This document is a deliverable of the AQUAEXCEL[2020], WP5/Joint Research Activity 1 – "Virtual laboratories and modelling tools for designing experiments in aquaculture research facilities".

Experiments with fish usually involve extensive use of laboratory facilities, and often need to run for long periods of time to obtain the desired answers. In such experiments, it is essential to ensure that the fish experience acceptable welfare conditions. Together with the potentially high costs of such experiments, the welfare perspective underlines the need for good experimental planning when doing research on fish. Both from an ethical (3R's) and a cost perspective, numerical models may be useful tools for experimental design, preparation and planning before realizing the actual experiments.

One of the main research activities in AQUAEXCEL[2020] WP5 is to develop a virtual laboratory system that enables virtual experiments in aquaculture research facilities. This system will feature a framework (see Bjørnson et al., 2016) that allows the integration of mathematical models of different subsystems in common simulations, replicating the system operation of research laboratories. The overall system concept as imagined in the start of the project is shown in Figure 1.

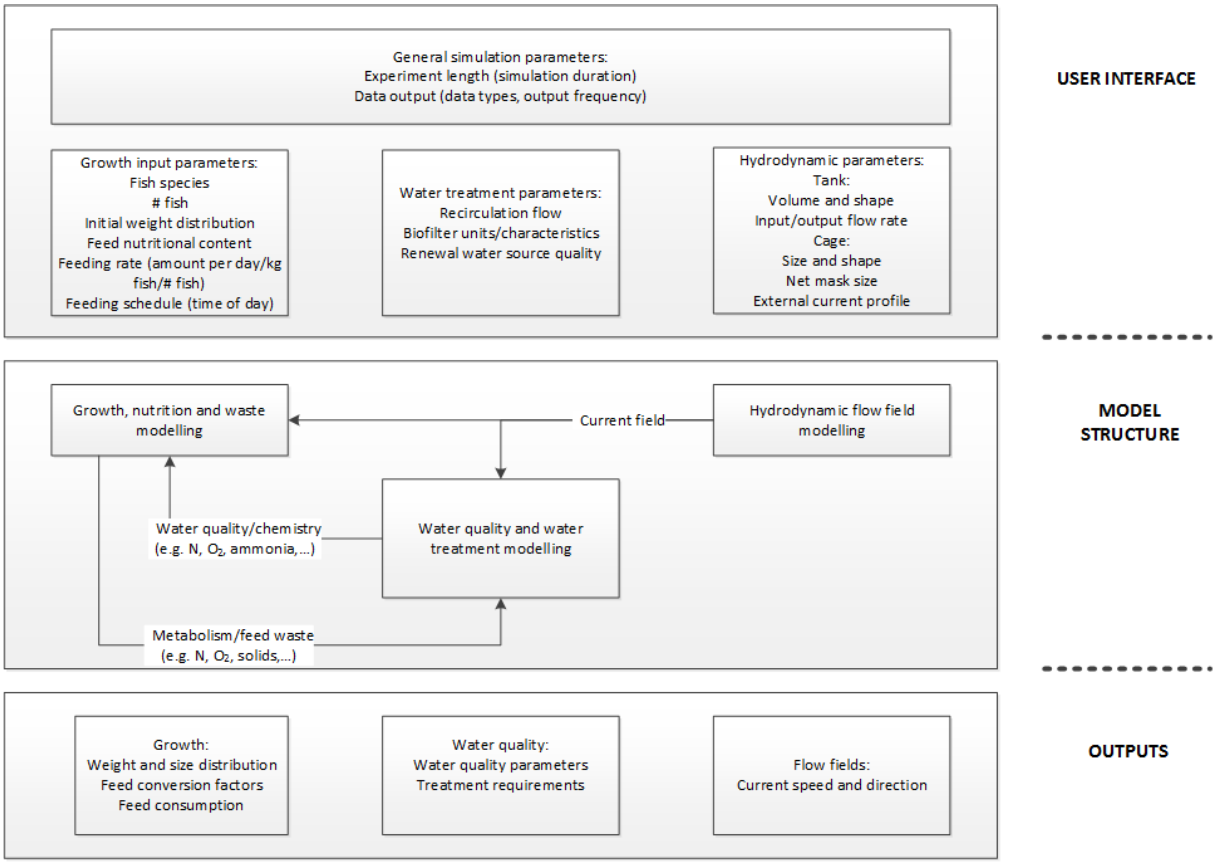


Figure 1 Virtual laboratory system concept

The purpose of this report is to present the second version of the virtual laboratory developed in WP5 of AQUAEXCEL[2020]. The first version of the virtual laboratory is described in Bjørnson et al. (2018). Since then, we have launched the website, done usability studies and improved on both the performance and usability of the site. For completeness, this report includes much of the background material from the previous report. For readers familiar with the previous report, much of the main components, framework and dataflow remains the same, but most chapters have received slight upgrades to reflect the new version. The major additions to this report are a subchapter on integration of the models in chapter 3, a subchapter on different FMU configurations in chapter 4, and a complete overhaul of chapter 5 to demonstrate the virtual laboratory.

The rest of this report is structured as follows: First, the general framework and system architecture used to integrate the different submodels is presented. This is followed by short descriptions of each submodel, covering their function and outputs, and some information on how they were implemented into the common framework as well as how they work together. We then present the implementation of the Virtual Laboratory Version 2, first from a technical viewpoint, then through a use case where a user conducts a new experiment. We then sum up the implementation of the virtual lab in the conclusion.

2. Framework and System Architecture

This section briefly presents the results reported in Bjørnson et al. (2016) describing the system architecture/technical framework that was chosen for developing virtual laboratory solution for the aquaculture domain.

The standard called Functional Mock-up Interface (FMI) defines how different simulation models realized in different simulation environments may be integrated in common simulations. Based on the extensive use of this standard in other industry segments (e.g. automotive and maritime industries), and the ability to handle models implemented in different systems/tools (see https://www.fmi-standard.org/downloads for a list of eligible tools), FMI was chosen as a basic framework for model integration in the AQUAEXCEL2020 virtual laboratories.

FMI defines an interface to be implemented by executables called Functional Mock-up Units (FMU) which contain the submodels. The FMI functions can then be used by a simulation environment to create one or more instances of an FMU and simulate them, typically together with other models. An FMU may contain its own solver, in which case it is possible to use *FMI for Co-Simulation*, where the submodels communicate by exchanging output values at each communication time step. Alternatively, if the FMU does not contain a solver, it is recommended to use *FMI for Model Exchange*, where the simulation environment connects two or more submodels to a common solver. Figure 2 illustrates the co-simulation model.

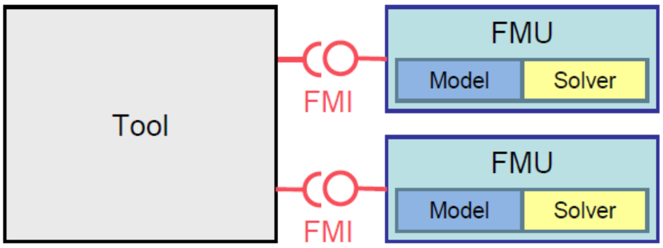


Figure 2 FMI for Co-Simulation (From Blochwitz 2014)

Irrespective of whether FMI for Co-Simulation or Model-Exchange is used, the integration between and execution of FMUs is governed by an FMI-master application. This application is responsible for synchronising the submodels at regular *communication time steps*, and for collecting all outputs and assigning all inputs of all FMUs in the system. The information flow between submodels that are to be interconnected through their respective inputs and outputs is thus maintained by the FMI-master algorithm rather than being realized as direct information exchange between the FMUs containing them.

As shown by Bjørnson et al. (2016), the overall initiation of models can be handled with the architecture described in Figure 3. The Master algorithm receives the experiment setup from the user interface and then asks FMU providers for the necessary FMUs (Stage 1). The FMU providers then handle the initiation and setup of the necessary FMUs and returns them to the Master which then executes the simulation (Stage 2).

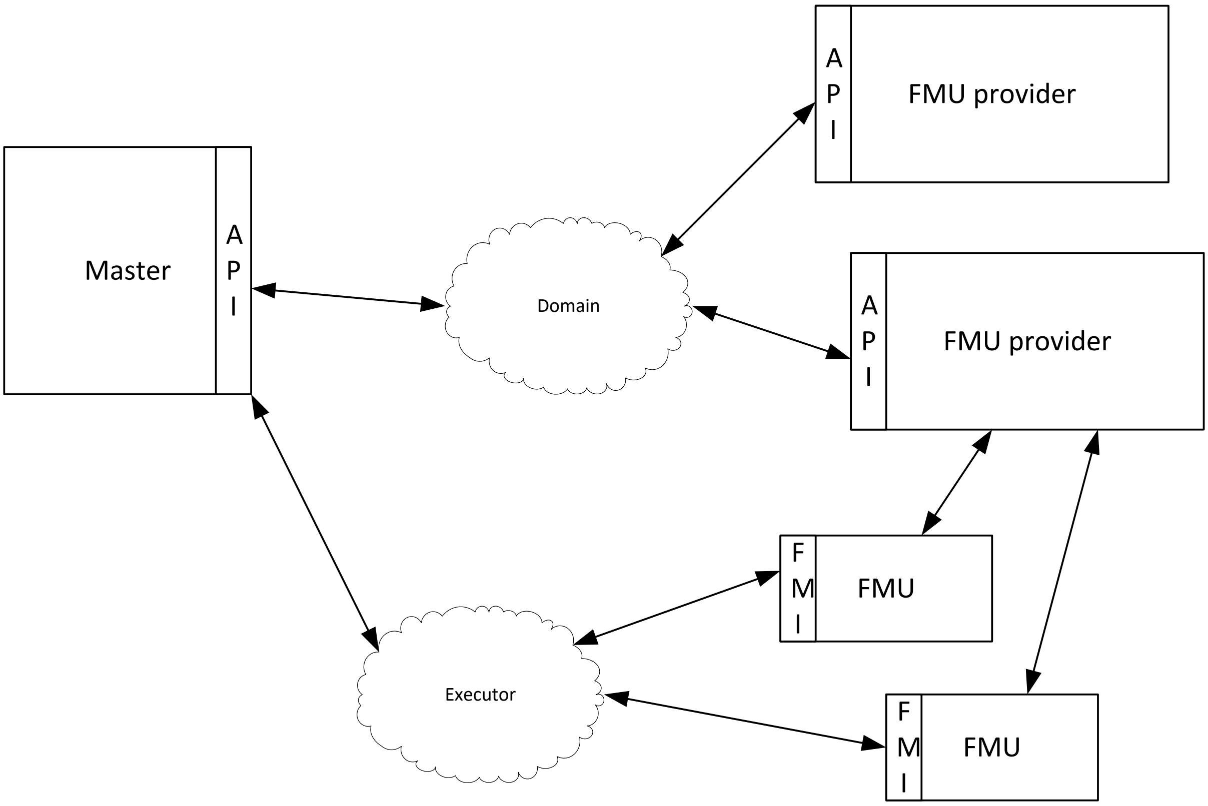


Figure 3 Master initiation architecture

The AQUAEXCEL^{2020} virtual laboratory is built upon an open-source FMI-master application, Coral, developed by SINTEF Ocean, NTNU and a consortium of industry participants from the maritime segment (see https://viproma.no/ for more information). At present, three different FMUs are possible to use in the virtual laboratory: simulating fish growth, water treatment and tank/cage hydrodynamics, respectively. In AQUAEXCEL^{2020} the aim is to primarily use FMI for co-simulation, and this entails that the FMUs developed for the virtual laboratory need to output their variable values at each communication time step.

Since the essential idea behind FMI is to facilitate integration of independent models in common simulations, future expansion and adaptation of the AQUAEXCEL^{2020} virtual laboratory to other types of experiments is an easy task. This can either be done by expanding the portfolio of eligible submodels with new FMUs or by modifying the existing FMUs. Furthermore, Bjørnson et al. (2016) observed that when using FMI, the ability to develop models using any programming language or modelling framework (as long as they appear in the list of eligible tools at https://www.fmi-standard.org/downloads), is a large advantage with respect to collaborative work between institutions. This also fits well with the format of WP5 in the AQUAEXCEL^{2020} project, where all submodels are to be developed by different partners. To ensure security, backup and versioning history of the code of all FMUs, SINTEF have provided the project partners with access to their code collaboration server code.sintef.no.

3. INTEGRATED MODELS

The following numerical models are the main components to be included in the virtual laboratory developed in WP5 of AQUAEXCEL^{2020}:

- Growth, nutrition and waste production models for different fish species (task 5.1.)
- Water quality and water treatment modelling (task 5.2.)
- Modelling of hydrodynamic flow fields in tanks and cages (task 5.3.)

These models have been realized as FMUs using the principle of Co-Simulation.

Detailed outlines, validation and discussions on the models delivered in tasks 5.1, 5.2 and 5.3 can be found in Lika, K. et al. (2020), Abbink, W. et al. (2020) and Alver, M. O. (2020), respectively. In this section, we give a short summary of these deliverables for completeness of this report. We also describe how they are integrated in the Virtual Laboratory.

Fish model

The AquaFishDEB model presented in Lika, K. et al. (2020) is designed to predict growth, feed consumption and waste production for Atlantic salmon, seabream and trout. Specifically, the model predicts 1) fish growth for different feeds (quantity and composition) and water temperature, and 2) oxygen consumption and waste production (nitrogen, CO_2, solids) at different fish sizes, temperatures, feed rations and diet compositions for individual fish or groups.

The model is based on the Dynamic Energy Budget (DEB) theory for metabolic organization, which provides a conceptual and quantitative framework to study the whole life cycle of individual animals while making explicit use of energy and mass balances (Lika, K. et al., 2020). The model covers all life stages of a fish (including larvae, juveniles and market size fish) and is explicitly tied with feed and temperature. It accommodates different feeding strategies (e.g. ad libitum or restricted, feeding frequency, adaptive feeding) and feed compositions. The output of the model includes fish growth characteristics (number of fish, mean body-size, total biomass, feed intake, specific growth rate and feed conversion efficiency), waste production (faecal dry matter and nitrogen-loss, as well as non faecal nitrogen loss) and gaseous exchange (O_2 consumption and CO_2 production).

As presented in Lika, K. et al. (2020), predictions made by the AquaFishDEB model are the end products of a two-step modeling procedure (Figure 4). The first step involves the parameterization of the DEB model for each species. In the second step, the DEB parameters are used in the prototype AquaFishDEB model that then simulates the dynamics for a group of fish exposed to user input regarding fish and feed characteristics, and the specified experimental conditions. The developed prototype model is thus able to predict growth, feed consumption and waste production for the fish.

The first step of the modeling procedure has been accomplished for three chosen species: rainbow trout, seabream and atlantic salmon.

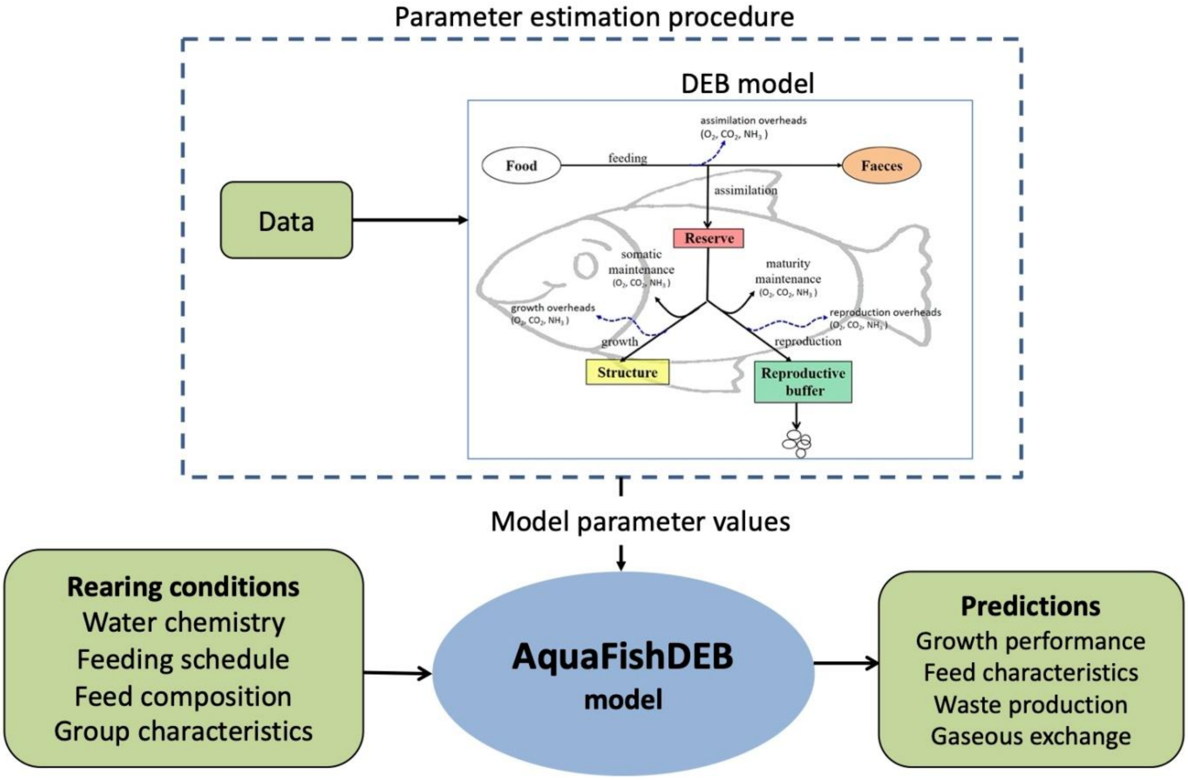


Figure 4 Schematic representation of the two-step procedure for the development of the AquaFishDEB model. (From Lika, K. et al. (2020))

The AquaFishDEB model was first implemented as a stand-alone model in Matlab, and all model tuning, verification and validation was done using this version of the model. After this, the Matlab code was converted into C++ using Matlab Coder, thus enabling simulations of the model in C++. This code was then linked into an FMU-interface implemented in C++, that was compiled, resulting in an FMU containing the AquaFishDEB-model. The functionality of this version of the model was finally verified by comparing model outputs from the FMU with those obtained with the original Matlab implementation.

Water treatment model

Abbink, W. et al. (2020) presents a model that predicts the water quality and water treatment effects in research infrastructures such as tanks. The model was designed as a generic tool that users of research facilities could use prior to the start of an experiment to predict the expected water quality during the experiment. In addition, the model could be a tool for (re-) designing systems so that they result in the desired water quality for the experiment envisioned. This makes the model a potential tool for teaching TNA users, research infrastructure technicians and others involved the principles of water quality control in fish culture units. The model uses input on waste production from task 5.1 as a starting point.

The sub-model computes water quality based on input parameters that describe the production plan and the experimental design. These inputs may either come from the growth model in task 5.1 or be provided as direct inputs to the model if they are known in advance. The model outputs describe water quality using the most crucial parameters related to ammonia and nitrate in the system (tanks and filters). For each communication time step, the model calculates values such as ammonia production by the fish, nitrification rate, nitrification capacity, ammonia load to the biofilter, ammonia removal rate, ammonia concentration in the water, nitrate production, and nitrate in the tanks. Figure 5 provides an overview of the major components of the model.

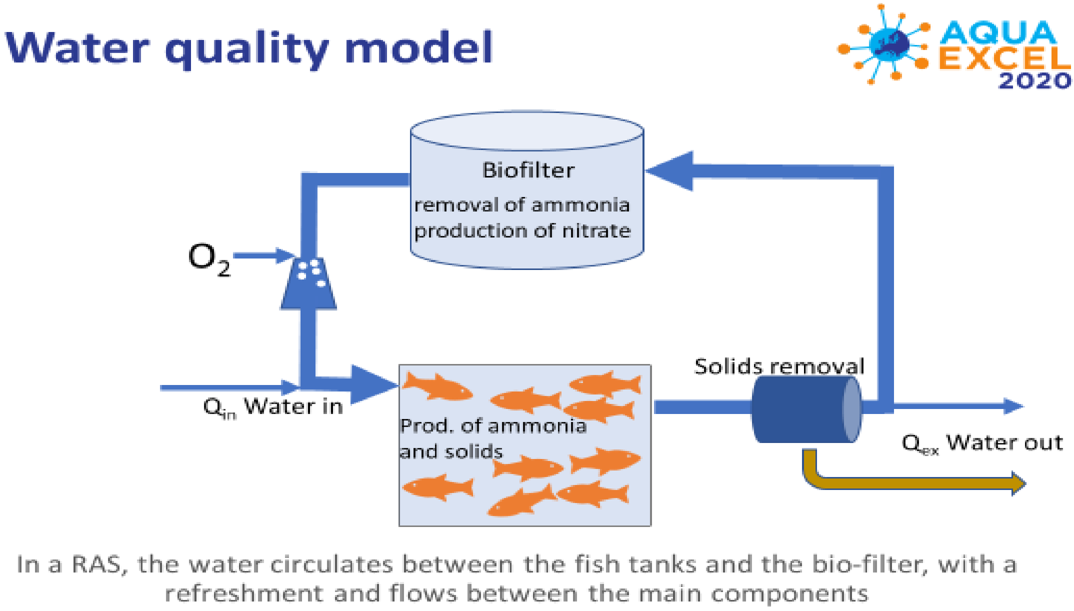


Figure 5 Water quality model (From Abbink et al. 2020)

The water quality model was first implemented in Excel, and all tests and tuning of the model was done using this implementation. To convert this model into an FMU, a C++-FMU-interface containing all necessary FMI functions was created. This interface communicated with the Excel implementation by linking in functionality provided by the LibreOffice API and containing a LibreOffice runtime environment. In conclusion, this resulted in an FMU that used the original implementation in Excel to simulate water quality and treatment, meaning that the FMU operated identically to the validated first implementation. For the final version of the water quality model, the model was fully reimplemented in C++ to enhance performance and remove the dependency on the LibreOffice API.

Hydrodynamic model

The objective of the flow field model presented by Alver, M. O. et al. (2020) was to represent the water currents within the production unit (fish cage or tank), presenting key information related to the current to the other model components. The developed flow field model uses one approach for current in tanks – precomputed flow fields from a CFD model (see Figure 6) – and another for open sea cages – current profiles depending on ambient current conditions. Currently three tanks are presimulated as well as four sea cage locations. The model interacts with the other model components either through providing the current speed and direction vector for given locations, or through providing descriptive numbers for the overall flow field in the production unit.

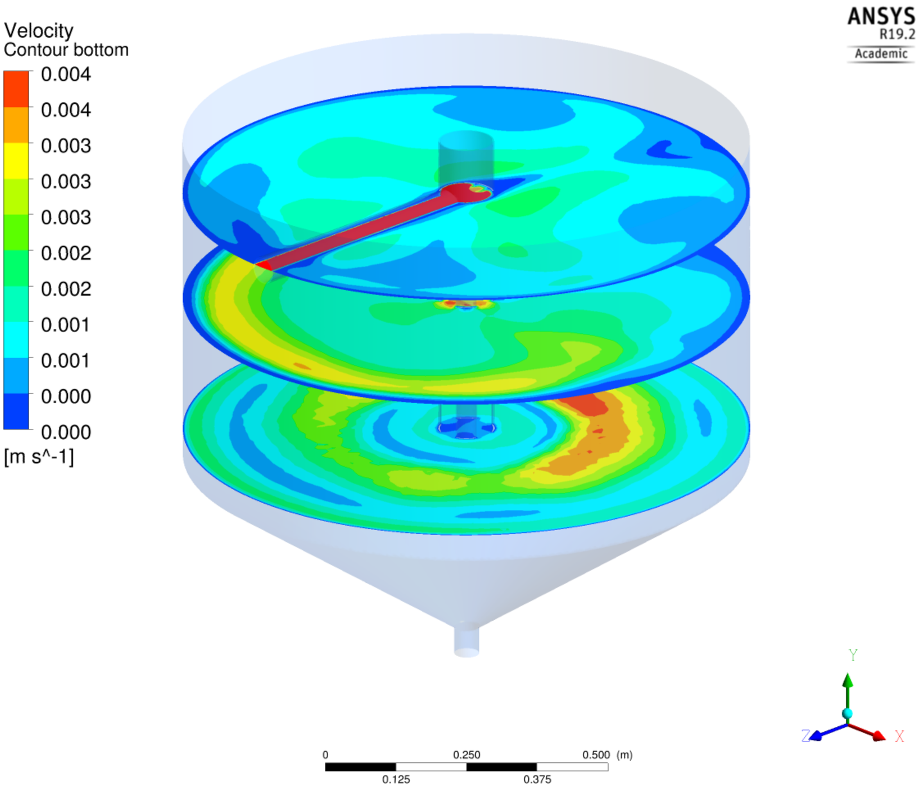


Figure 6 Example of velocity flow field in HCMR tank, computed using Ansys Fluent. (From Alver 2020)

The flow field model component is written in C++ and includes a similar FMU-interface as the FMUs for fish growth and water quality. Internal mechanisms for reading outputs from CFD-simulations were programmed directly in the C++ implementation. By reviewing the outputs from the FMU against the outputs from CFD it was possible to ensure that the data were properly read.

Integration

Figure 7 shows the interconnections between the different submodels. Input parameters are shown above the dotted line in blue, if the parameters are imported from a different FMU they are shown in yellow. If the selection in another FMU precludes changing a parameter it is shown in red. Output variables are shown below the dotted line in blue, if they are used by another FMU they are shown in green. If there is a direct connection from an FMU to another it is shown with a line, if there is an indirect connection it is shown as a dotted line.

The most important connection is between the DebGrowth and WaterTreatment FMU. DebGrowth calculates the number of fish, their medium wet weight, feed intake, total oxygen consumption, total CO2 production, total nitrate production and total production of dry matter for each timestep. These variables are sent to the WaterTreatment FMU as continuous inputs for each timestep.

Environmental variables such as temperature, oxygen, salinity and pH are assumed to be regulated to remain constant during the experiment. So if for example an experiment is set up where the oxygen level would drop beyond the threshold of fish survivability the experiment will still run to its conclusion assuming enough oxygen for the fish to survive, but the user will be able to see when oxygen input to the tank needs to be increased in order to stay within acceptable limits.

The flow field FMU currently has a more indirect impact on the other FMUs. If a pregenerated flow field is selected for the experiment, the parameters relating to tank size and flow rate are preselected in the WaterTreatment FMU. The connected recirculation equipment can still be configured for a different setup. If an open cage is selected, the WaterTreatment FMU is disconnected from the simulation. All the FMUs deliver their output variables to the Reporter FMU which handles the storage of results for later presentation.

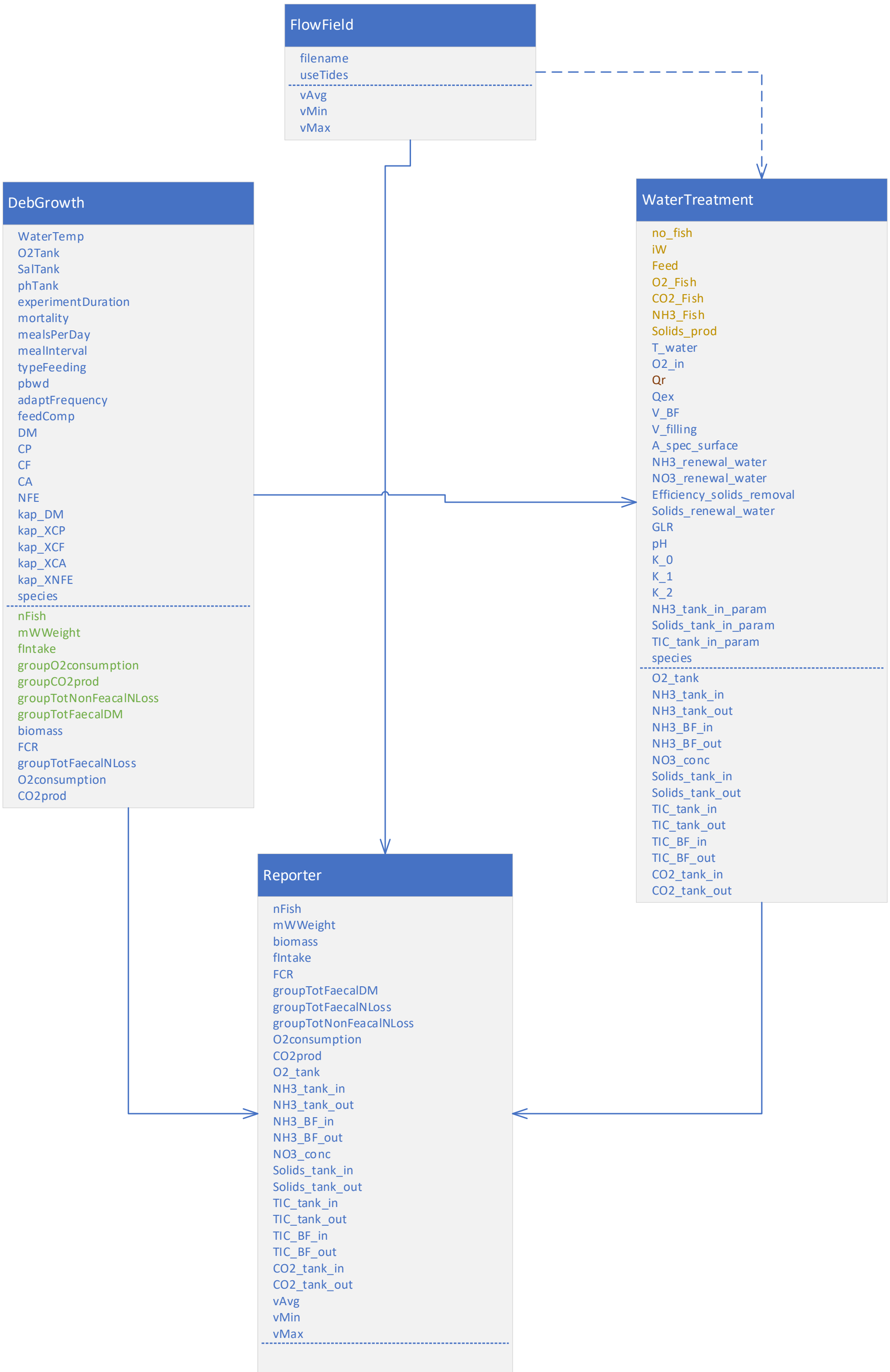


Figure 7 FMU Connections

4. VIRTUAL LABORATORY VERSION 2

To describe the Virtual Laboratory Version 2, we will look at the software from different perspectives. First, we look at the main components and data packages, and then we consider the viewpoint of a user and look at the user interface flow. Afterwards, we take a high level technical view on communication and the data model, before going into details on the data flow for a simulation. We also provide an overview of different FMU configurations.

Main components and data packages

There are three main components of the virtual laboratory: The web interface, built on Django[1] technology, the FMUs from the sub-workpackages built on different technologies but all providing a C interface, and Coral[2], an open source co-simulation software with support for FMI, built and maintained by SINTEF Ocean.

For details on the FMU implementations we refer to section 3. For more information on Coral we refer to the online documentation[2]. This section will focus on the implementation of the user interface and the integration between the components to realize a complete simulation.

The user interface is implemented in Django, which in turn is running in a Python environment. Django is compatible with a multitude of databases, and in our case, we have chosen to run a MySQL database for the project. To separate functionality of the user interface, we have separated functionality into different applications running on a common project environment. Figure 8 provides an overview of the data packages currently running on the system. The Users application handles authentication and personalization of users. The Simulation application is responsible for setting up and running simulations. The Results application handles everything related to showing results of the simulations to the user. And, finally, the Admin application provides super-users with an interface to set user privileges.

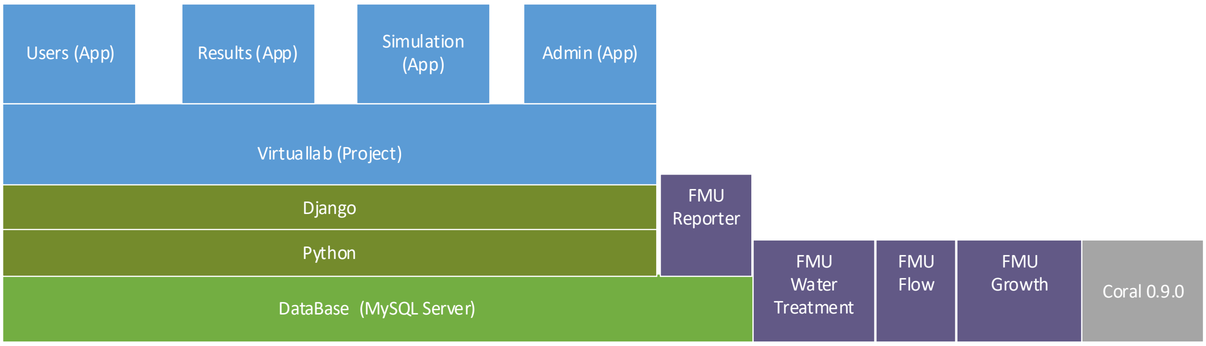


Figure 8 Overview of data packages

User interface flow

Figure 9 provides an overview of the user interface flow. For the user authentication, we are using Django's built in authentication system. Thus, we route any access to the site to the Home page if a user is not authenticated. An anonymous user has few options at the site, limited to logging in, requesting a new password or registering a new user. As an added security measure, a new user needs to be authenticated by an administrator before gaining access to the rest of the site. This second level of authentication can be set to manual or automatic depending on how much load the site is experiencing. For example, when the site

[1] https://www.djangoproject.com/
[2] https://github.com/viproma/coral/

is being used in a course or presented to industry it might be beneficial to set authentication to automatic.

Once registered and authenticated a user has several more options. They are first routed to the Home page for authenticated users which provides more information about the virtual laboratory. From there the user can choose to log out which returns them to the anonymous Home page. In addition, they can access their user profile, where they can change their information or password, review the privacy policy or delete their profile. They can also access more information about the different models currently implemented in the virtual laboratory, how they work together, and read a short howto guide on how to set up and run experiments. The main point of interest will, however, be the Simulate and Results pages. These pages provide the user with their personal configured experiments and results thereof.

The Simulate page provides access to all previous experiments the user has run, as well as access to a simulation wizard to configure new experiments. We have made the constraint on the system that the user is only allowed to simulate a growth model together with a water quality model, as well as an optional hydrodynamic model. The user specifies the infrastructure setup together with the biomass and feeding regime, before selecting the length of the experiment. All parameters are validated in web forms before the simulation starts and the user is returned to the start of the Simulation page.

Once a simulation has finished the experiment results will be available at the Results page. The user can choose which experiment and which parameter to review, as well as downloading the complete dataset in csv format for further analysis offline.

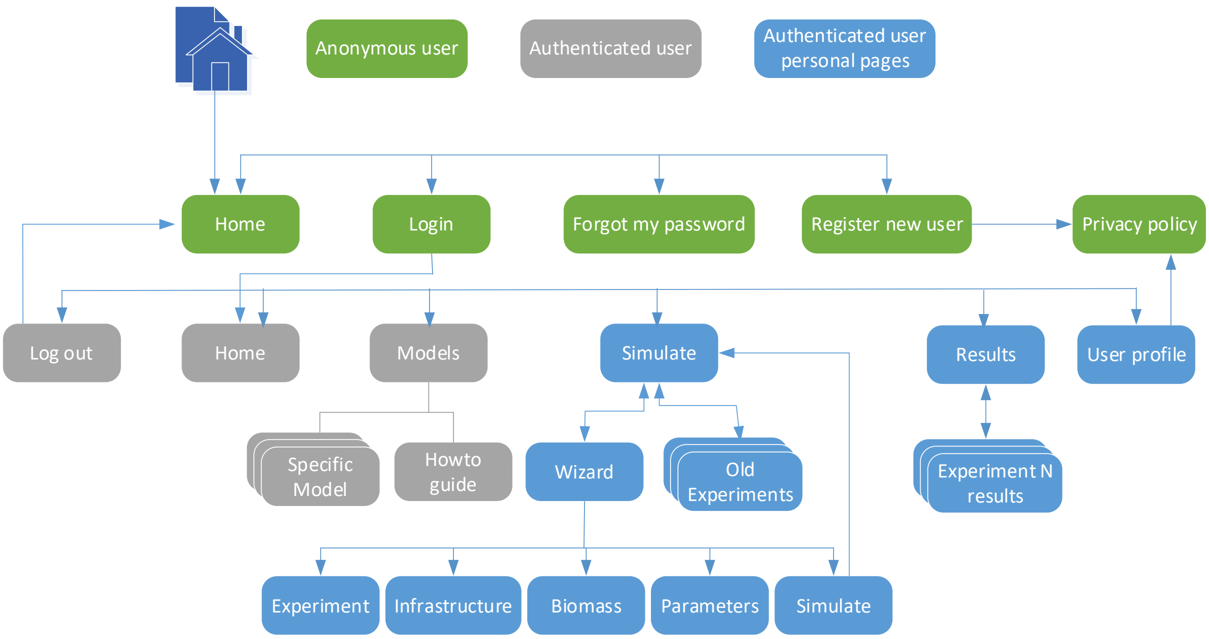


Figure 9 Web structure

High level communication

Figure 10 describes the overall flow of information in the Virtual Laboratory. To understand the flow, we need to go into the communication flow of the Django framework we are building upon. The Django framework closely follows the Model View Controller Architecture[3]. However, since the Control part is covered by the framework, and most of the action happens in the views and template layer, it is often referred to as a Model Template View architecture. The Model layer handles everything related to the data: access, validation, behavior and relationships. The

[3] https://en.wikipedia.org/wiki/Model-view-controller

template layer contains presentation logic, how content should be presented to the user through Web pages or other types of documents. The View layer contains the business logic, it functions as a bridge between the data in the models and the presentations in the templates.

Communication between the Model layer and the database is abstracted away in the Django framework. We only need to access the data in the Model layer and the underlying framework will update the database for us. Functions in the View layer has full access to variables and methods in the Model layer. Data from the View may be passed to appropriate templates which are then rendered to be presented for the user in a browser. The user then provides inputs which are transferred through the URL dispatcher back into an appropriate View.

To improve the user experience of the Virtual Laboratory, we make use of Bootstrap[4], an open source toolkit for developing with HTML, CSS and JS, as well as Highcharts[5] an SVG-based, multi-platform charting library for presenting the data in the template layer. In order to process the data at the View layer, we make use of Pandas[6], an open source library providing data structures and data analysis tools for Python.

To enable co-simulation, we use the Coral package. It is initiated from the View layer and updates the virtual laboratory through a Reporter FMU which writes results back to the database. We will examine this process in more detail in the Detailed data flow section.

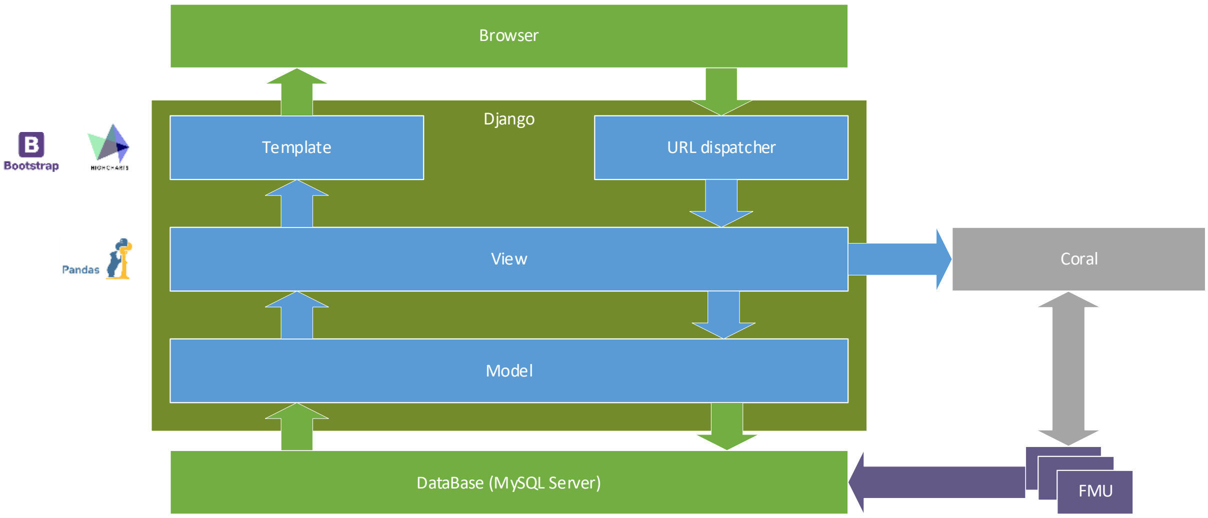


Figure 10 High level communication

Data structures

Data structures for the virtual laboratory is handled by the Model layer and mirrored in the MySQL database (see Figure 11). A data structure in Django is called a Model, not to be confused by the simulation models implemented in FMUs. A Model is mirrored in a table in the database. The naming convention of the data table is "ApplicationName_AreaName". All models have access to their related models even if they are defined in different applications. An overall view of the data structures and their relationship in the database is described in Figure 11 in crowfoot notation.

At the core of the data structure is the experiment, defined in Simulation_experiment, this data model contains information on whether the experiment has been validated, if it has started, and what the progress of the simulation is. A user defined in User_customuser can link to multiple experiments. For the Version 2 of the virtual laboratory an experiment has an

[4] https://getbootstrap.com/
[5] https://www.highcharts.com/
[6] https://pandas.pydata.org/

infrastructure model defined in Simulation_infrastructure, a biomass model defined in Simulation_biomass and a list of core parameters defined in Simulation_experimentcore.

Simulation_infrastructure contains the input parameters for the watertreatment and hydrodynamic model, as well as some input parameters for the growth model. Simulation_biomass contains the input parameters for the growth model along with information on validation ranges. Simulation_experimentcore contains information on the start and stop time for the simulation as well as the integration step size. Once run the results of the simulation is stored in the Results_simulationsresults table.

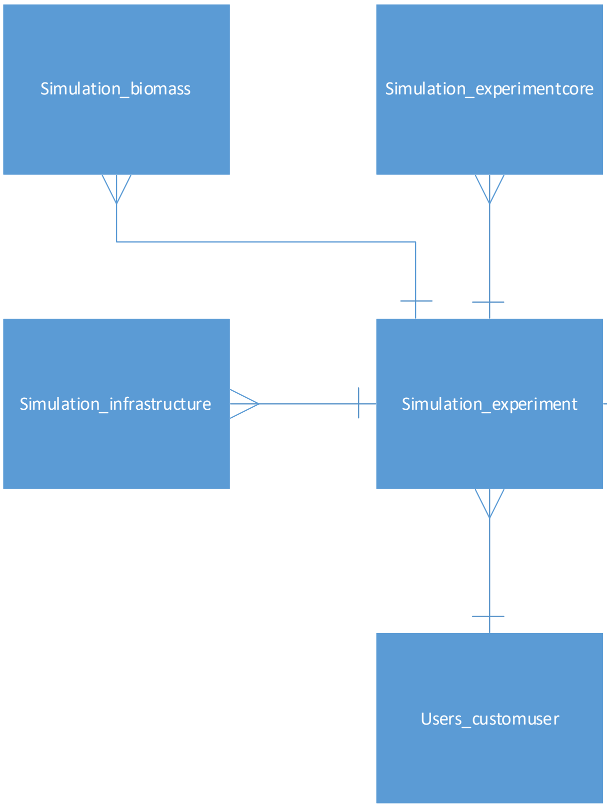


Figure 11 High level data structure in crowfoot notation

Detailed data flow

The main challenge of the Virtual Laboratory has been to achieve a seamless integration with the underlying co-simulation software Coral and the FMUs provided by the sub-workpackages. In this section, we delve deeper into our solution for integration of the underlying simulation software. Figure 12 provides an overview of the detailed data flow which can be divided into three different stages we go through these stages in detail below.

The first stage of integration is using Corals integrated SlaveProvider and registering the FMUs with the provider as shown in Figure 12 Stage 1. This makes the FMUs available on the localhost for simulation.

The second stage takes place within the webframework of the virtual laboratory (Figure 12 Stage 2). A user registers a new experiment and validates the input data, this happens through Django's integrated form validation procedures. Once all input has been validated and set in the Model layer, the user gains access to the Simulate button which activates the simulation.

In the third stage, a call is sent to the underlying experiment model which generates an execonf and sysconf file for the experiment based on the parameters set in the Model layer. For details on what goes into these files see Bjørnson et al. (2016). These files are then sent to the CoralMaster software for simulation as illustrated in Figure 12 Stage 3. The CoralMaster checks the localhost for available FMUs. The CoralSlaveProvider responds with available FMUs and if these matches the FMUs required by the CoralMaster, new slave instances of the FMUs are activated for the CoralMaster. The CoralMaster then administers the co-simulation, providing a databus for data exchange between the CoralSlaves at each integration step. The CoralSlaves handles simulation between integration steps and queries and delivers data at the integration steps. In order to feed the data back into the virtual laboratory, we make use of a Reporter FMU which is tasked with reading parameter data at each integration step and feeding that data directly into the database.

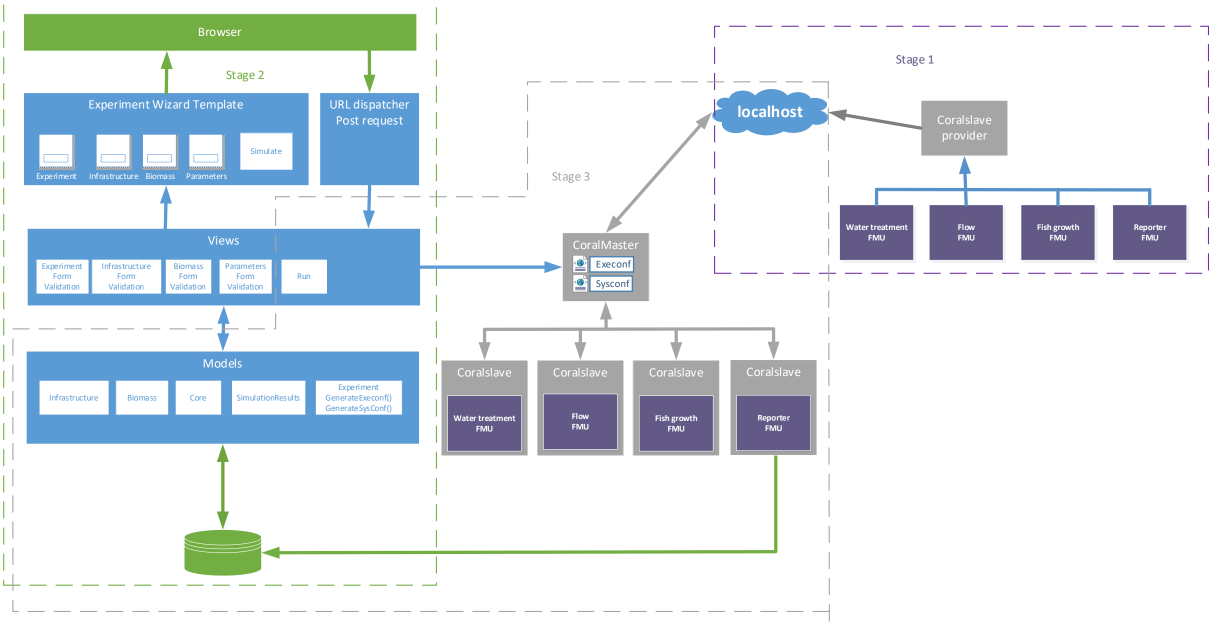


Figure 12 Detailed data flow for a simulation

Different FMU Configurations

The Virtual Laboratory version 2 has three main ways of configuring the four available FMUs depending on the user's choice of infrastructure for the experiment.

In configuration A (Figure 13) the user has chosen a presimulated tank for the experiment. The growth FMU delivers data to the watertreatment model and the reporter, and the watertreatment and flow FMUs delivers data directly to the reporter. This setup will lock certain input parameters in the watertreatment model, since changing them would result in a different flow than the presimulated one. These parameters will be removed from the user interface, so they are not changed in the experiment setup.

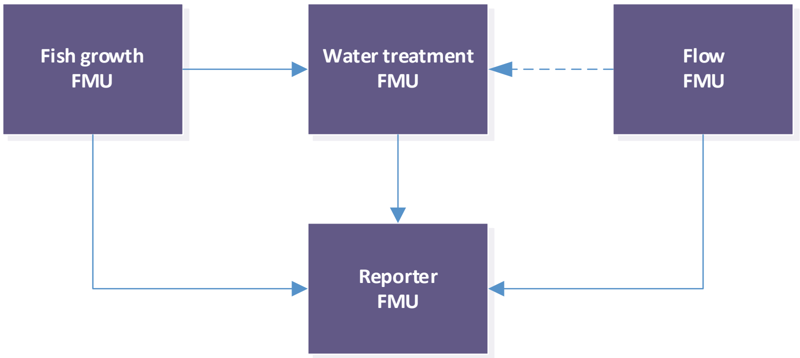


Figure 13 FMU Configuration A

In configuration B (Figure 14) The user has chosen a custom tank. Since the flow model is depending on presimulating tanks due to processing capabilities, this disconnects the flow FMU from the setup, but otherwise it remains the same as in configuration A.

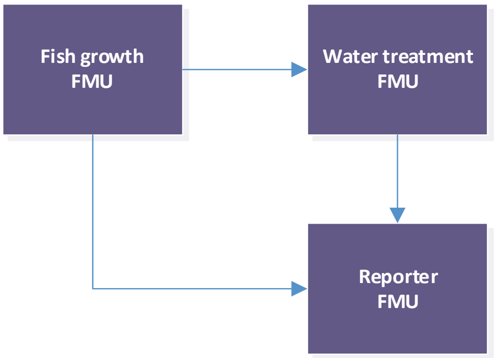


Figure 14 FMU Configuration B

In configuration C (Figure 15) The user has chosen a netcage as the infrastructure. Since the watertreatment is only necessary for tank simulations, it is disconnected from the simulation, and the growth and flow FMUs run in parallel, delivering data to the reporter FMU.

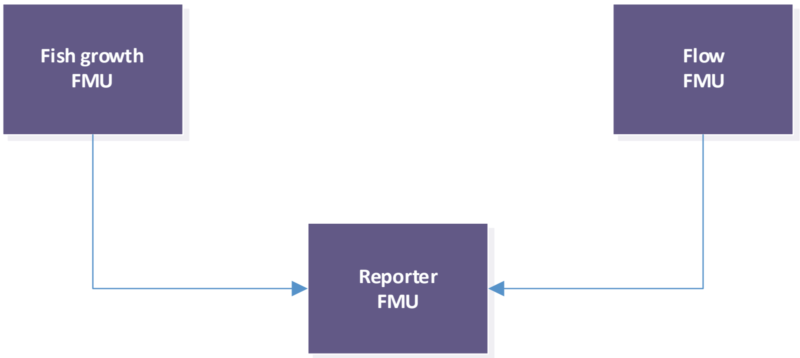


Figure 15 FMU Configuration C

Direct framework access

The webframework of the Virtual Laboratory Version 2 provides an easy to use access for users who want to simulate experiments. However, the trade-off we make for increased usability is a decrease in flexibility.

The underlying framework does not have any limitation on experiment setup however, so it is possible for advanced users to configure more elaborate experiments. This is done by bypassing the webframework as illustrated in Figure 16. The loading of FMUs by the CoralSlaveProvider is the same as described above, but Stage 2 is bypassed by specifying an experiment execonf and sysconf file. These configuration files are then sent directly to the CoralMaster in Stage 3. Results of the experiment is then available as csv files, and the user needs to post-process the files themselves.

In order to take this approach, a user needs insight into each of the underlying model FMUs and how they might be connected, this information can be obtained from the *modeldescription.xml* file included in all FMUs. In addition, they need to setup their own Coral environment and know how to specify their own configuration files. More information on how to do this is available at https://github.com/viproma/coral. This might be beyond most common users of the system, but it allows project participants to experiment with advanced simulations.

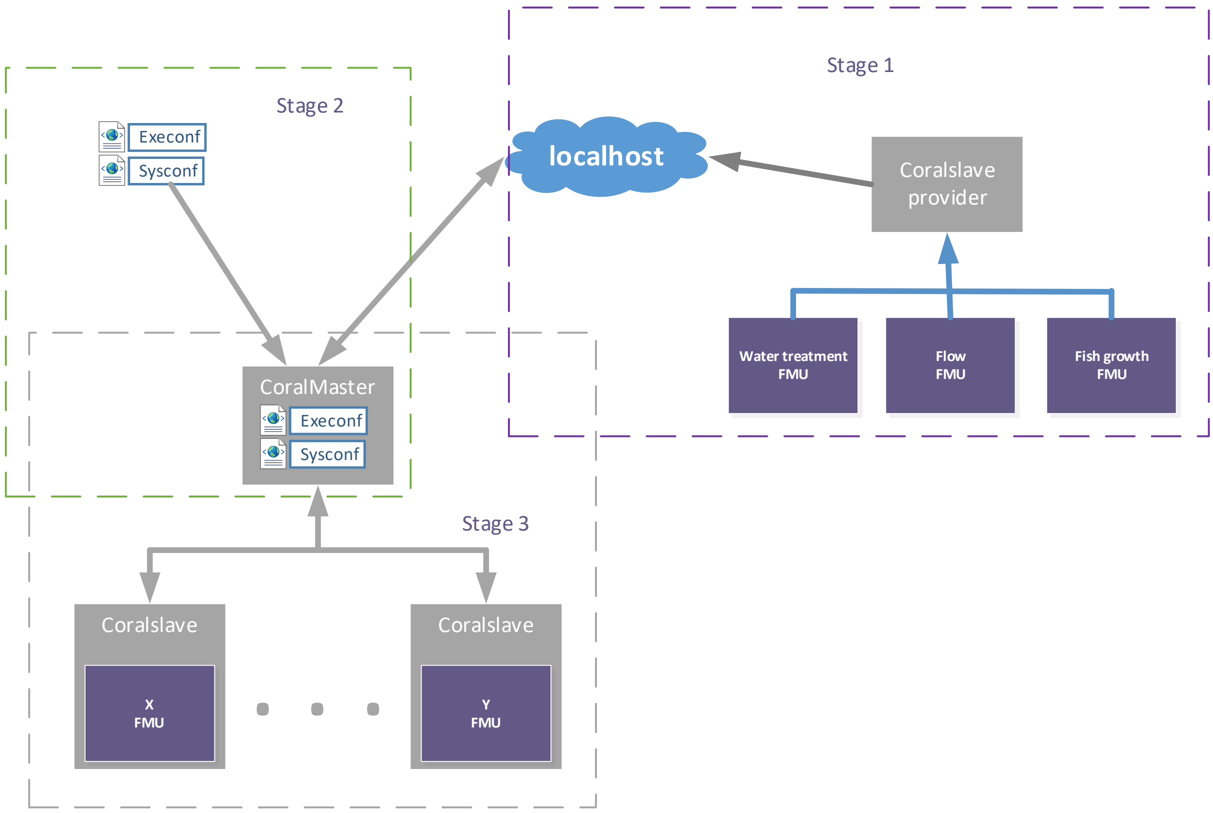


Figure 16 Direct Framework Access

5. DEMONSTRATION

In this section, we demonstrate the user interface of the Virtual Laboratory. First, we provide a walkthrough of the static information pages and what information can be expected to be found in each one. We then configure a simple experiment and review the results.

Figure 17 is the first page that greets the user after logging in. It contains information on the AQUAEXCEL2020 project and more details on the workpackage the Virtual Laboratory was developed in. The right side of the screen is reserved for news relating to updates of the Virtual Laboratory.

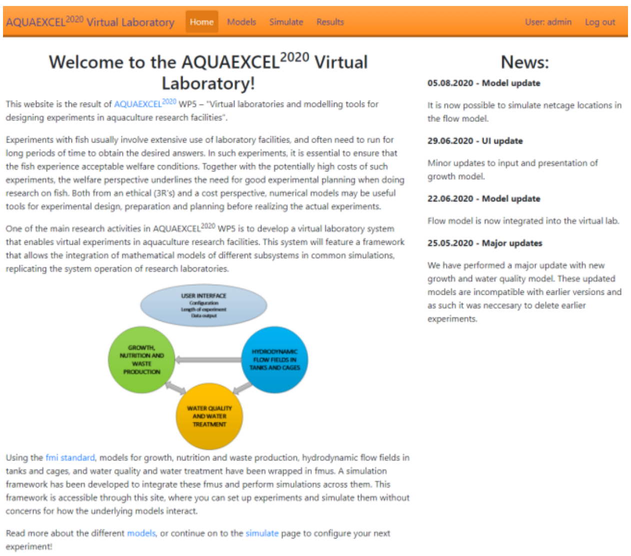


Figure 17 Welcome to the VL

From the menu line, the user can access their User Profile by clicking on their username, see Figure 18. Here, the user can review and update their information including changing their password, review the Privacy Policy or delete their profile.

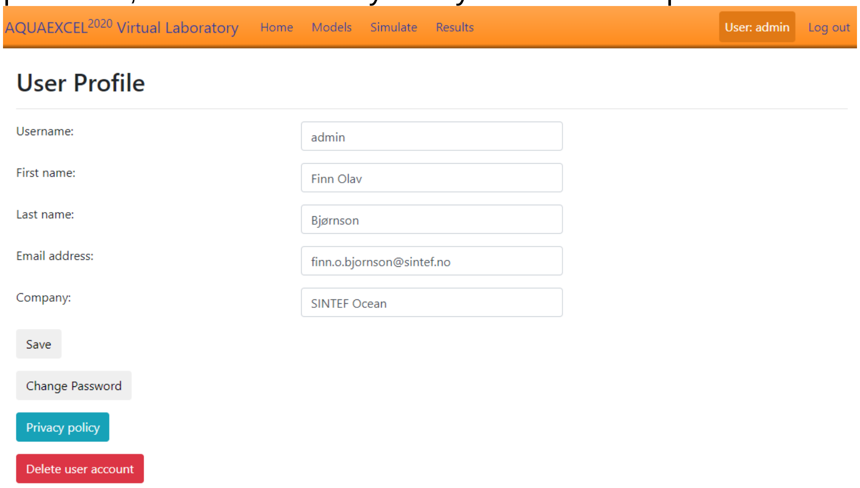


Figure 18 User Profile

The website provides some static information relating to the models, so the user can get a better understanding of how the underlying models work together when running an experiment. The first page is accessible by clicking the Models button in the menu line, see Figure 19. This page provides an overview of how the models work together, depending on the user's configuration in the Simulation.

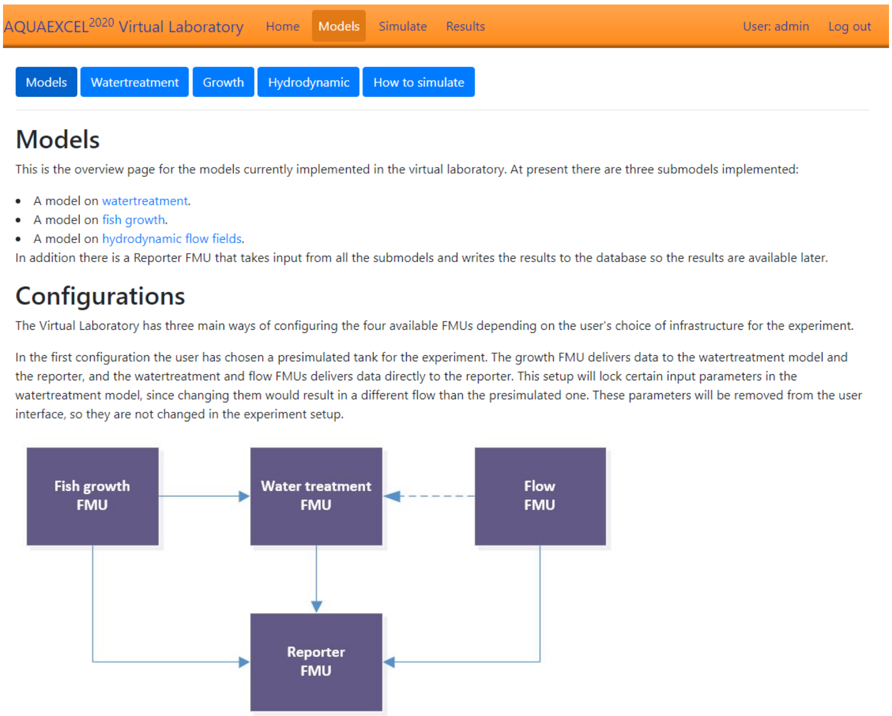
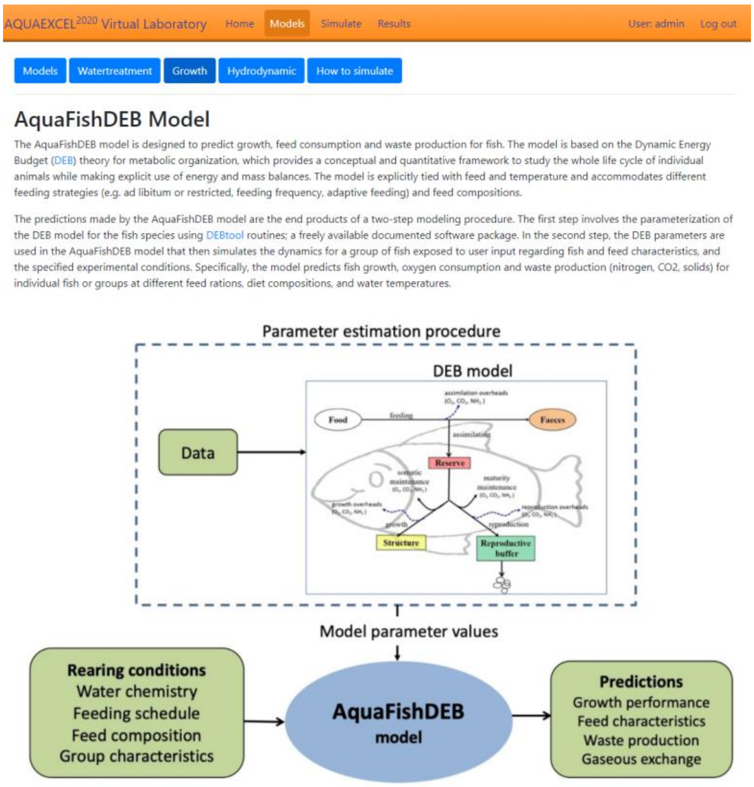


Figure 19 Model overview

Each of the implemented models have descriptive pages, accessible from the initial Models page. Figure 20 provides an example of the Growth model. Each submodel page contains information on the model itself with links to the official AQUAEXCEL2020 reports for more information.


AQUAEXCEL2020 Virtual Laboratory Home Models Simulate Results User: admin Log out

Models Watertreatment Growth Hydrodynamic How to simulate

AquaFishDEB Model

The AquaFishDEB model is designed to predict growth, feed consumption and waste production for fish. The model is based on the Dynamic Energy Budget (DEB) theory for metabolic organization, which provides a conceptual and quantitative framework to study the whole life cycle of individual animals while making explicit use of energy and mass balances. The model is explicitly tied with feed and temperature and accommodates different feeding strategies (e.g. ad libitum or restricted, feeding frequency, adaptive feeding) and feed compositions.

The predictions made by the AquaFishDEB model are the end products of a two-step modeling procedure. The first step involves the parameterization of the DEB model for the fish species using DEBtool routines; a freely available documented software package. In the second step, the DEB parameters are used in the AquaFishDEB model that then simulates the dynamics for a group of fish exposed to user input regarding fish and feed characteristics, and the specified experimental conditions. Specifically, the model predicts fish growth, oxygen consumption and waste production (nitrogen, CO2, solids) for individual fish or groups at different feed rations, diet compositions, and water temperatures.

Figure 20 Growth model

In addition to information on the model itself, the pages contain information on the inputs and outputs of the individual models. Figure 21 provides an example of some of the inputs and outputs of the watertreatment model.


AQUAEXCEL2020 Virtual Laboratory Home Models Simulate Results User: admin Log out

System characeristics	Total water volume fish tanks	L	
	Total water volume in the remaining part of the system	L	
	Recirculation water flow rate	L/d	
	System water exchange flow rate	L/d	
	Total biofilter volume	m^3	
	Filling percentage	%	
	Specific Surface Area (SSA) biofilter media	m^2/m^3	
	Removal efficiency drum filter	%	
	Gas:Liquid ratio biofilter (GLR)	-	
	Acid-base equilibria carbonate system (K0, K1, K2)		
Initial model values	Oxygen concentration fish tank in	mg/L	
	Total ammonia nitrogen concentration fish tank in	mg/L	
	Solids concentration fish tank in	mg/L	
	Total inorganic carbon concentration fish tank in	mg CO_2/L	

Output parameters

The water treatment model calculates the water quality for the fish tank inlets and outlets as well as the inlets and outlets for the biofilter in the system.

Description	Parameter	Units	Comment
Oxygen	Oxygen fish tank	mg/L	
Ammonia and Nitrate	Ammonia fish tank in	mg/L	
	Ammonia fish tank out	mg/L	
	Ammonia biofilter in	mg/L	
	Ammonia biofilter out	mg/L	
	Nitrate concentration system	mg/L	
Solids	Solids concentration fish tank in	mg/L	
	Solids concentration fish tank out	mg/L	

Figure 21 Water treatment model

The hydrodynamic model is special since it depends on presimulating tanks to a steady state. Figure 22 shows an example of what the users can chose from when selecting a tank for their experiment.

AQUAEXCEL2020 Virtual Laboratory Home Models Simulate Results User: admin Log out

Presimulated tanks

The following tanks have been simulated with the given parameters to a steady state and can be selected in the experiment wizard. It is also possible to configure a custom tank, but since the flow field model requires a lot of computational time to achieve a steady state this will result in flow data being unavailable.

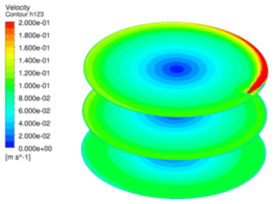
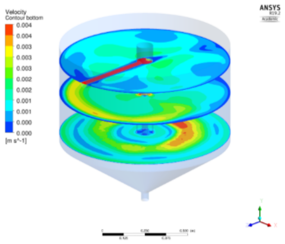
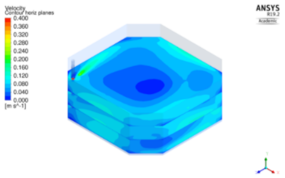
Tank	Type	Width (m)	Height (m)	Volume (l)	Flow rate (l/h)	Fluent simulation
VURH	Circular flat bottom tank	1.465	0.84	1400	1800	
HCMR	Cylindroconical tank	1.01	0.93	545	250	
Nofima	Octagonal exp. tank	1.8/1.75	1.0	3000	3600	

Figure 22 Hydrodynamic model

Finally, we provide a complete description of how to configure an experiment so new users have an easy entry into starting their simulations, see Figure 23.

AQUAEXCEL2020 Virtual Laboratory Home Models Simulate Results User: admin Log out

Models Watertreatment Growth Hydrodynamic How to simulate

How to simulate

Click the Simulate button on the menu line, this will take you to the simulation page where you have a choice of configuring a new experiment or revisiting your old experiments.

Creating a new experiment

To configure your experiment click the "Experiment Wizard" button, this takes you to the wizard which will guide you through the configuration procedure. In order to simplify the process of configuring an experiment, we have abstracted the underlying models away. Instead we have replaced it with a five step procedure where you validate your input. The underlying system will then take care of setting the correct parameters, initializing and connecting the models.

Naming your experiment

The first step of the process is naming your experiment, give it a name and click the "Validate" button. This name is the one that will be available on the previous experiments page, if you later wish to review your previous experiments. If the name is validated you are taken to the "Biomass" page.

Setting up your biomass

The second step is setting up the biomass for your experiment. You select the species you want, the initial number of fish for the experiment and their initial mean weight. Mortality is set as a percentage for the entire experiment period, and your population will decrease linearly towards the end of the experiment to achieve this predefined variable. Next specify the number of meals per day, and the interval between the meals. It is only possible to specify the interval in whole hours.

Then you need to specify the type of feeding. Ad libitum assumes enough feed is given to ensure all fish are full. Fixed feeding will feed each fish with a certain percentage of the initial body weight. You can specify this percentage if you choose this option. Adapted feeding is similar to fixed feeding but the percentage is calculated off an updated weighing of the fish to ensure more feed as the fish grows. The frequency of updates to the weight needs to be specified if this option is chosen.

Finally you need to specify the feed composition. You can chose to use standard feed, this will use FAO standards based on the production stage. Your population of Guilthead seabream and rainbow trout is categorized as fry up to 5 grams, fingerlings from 5 to 50 grams and production from 50 grams. Atlantic salmon is categorized as fry up to 5 grams, fingerlings from 5 to 100 grams and production from 100 grams.

If you do not wish to use standard feed, you can specify your own. You will then need to specify the feed in Dry matter, crude protein, crude fat, and nitrogen free extract. You will also need to specify the apparent nutrient digestibility for these parameters

Once you are satisfied with your biomass, click the validate button and if all values validates it will take you to the "Infrastructure" page.

Figure 23 How to simulate

The major part of activity in the Virtual Laboratory will take place on the Simulation page. Figure 24 shows the main page of the simulation area. The user may choose to review old experiments or launch a new one using the Experiment Wizard button. For our demonstration, we will continue to the Wizard.

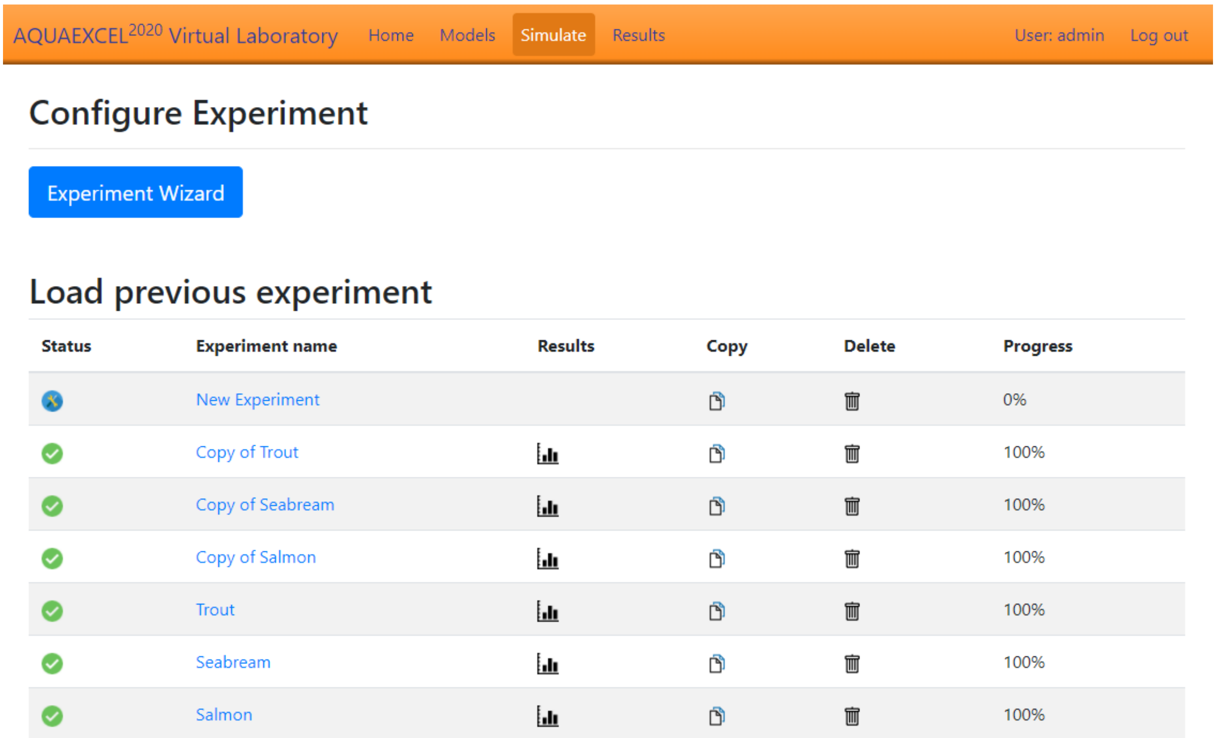


Figure 24 Simulation main page

Figure 25 shows the page that loads when the user starts the simulation wizard. The wizard is split into five subpages: *Experiment*, Biomass, *Infrastructure*, *Parameters* and *Simulate*. The user is first sent to the Experiment page where they may choose a name for the new experiment. This is the name that will be linked in the main simulation page under Previous Experiments, see Figure 24.

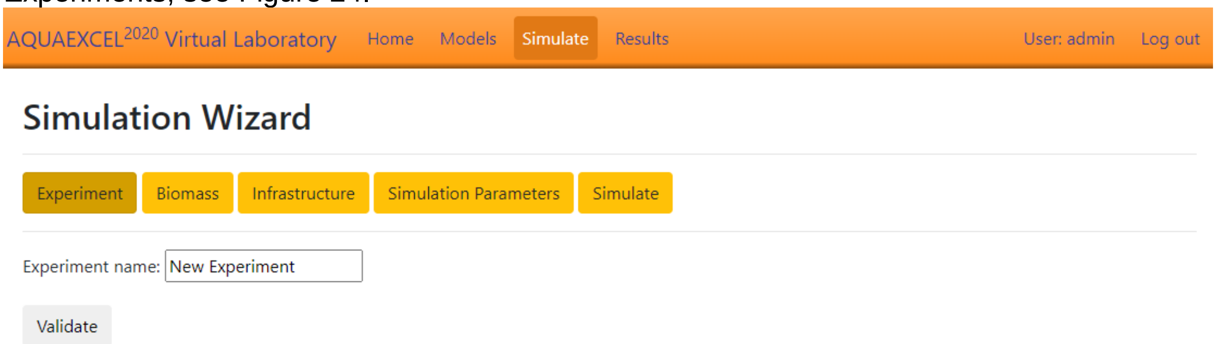


Figure 25 Simulation Wizard

All subpages of the wizard have a validation state. Once the user has confirmed the input, the Virtual Laboratory will validate if the parameters are within acceptable ranges and if they are turn the button from yellow (unvalidated) to green (validated). Figure 26 demonstrates this where the user has validated a new experiment name and is currently configuring the biomass.

To set up your biomass you select the species you want (Atlantic salmon, seabream or rainbow trout), the initial number of fish for the experiment and their initial mean weight. Mortality is set as a percentage for the entire experiment period, and your population will decrease linearly towards the end of the experiment to achieve this predefined variable. Next specify the number of meals per day, and the interval between the meals. It is only possible to specify the interval in whole hours.

Then you need to specify the type of feeding. Ad libitum assumes enough feed is given to ensure all fish are full. Fixed feeding will feed each fish with a certain percentage of the initial body weight. You can specify this percentage if you choose this option. Adapted feeding is similar to fixed feeding but the percentage is calculated off an updated weighing of the fish to ensure more feed as the fish grows. The frequency of updates to the weight needs to be specified if this option is chosen.

Finally you need to specify the feed composition. You can choose to use standard feed, this will use FAO standards based on the production stage. Your population of Gilthead seabream and rainbow trout is categorized as fry up to 5 grams, fingerlings from 5 to 50 grams and production from 50 grams. Atlantic salmon is categorized as fry up to 5 grams, fingerlings from 5 to 100 grams and production from 100 grams.

If you do not wish to use standard feed, you can specify your own. You will then need to specify the feed in Dry matter, crude protein, crude fat, and nitrogen free extract. You will also need to specify the apparent nutrient digestibility for these parameters.

Once you are satisfied with your biomass, click the validate button and if all values validate it will take you to the "Infrastructure" page.

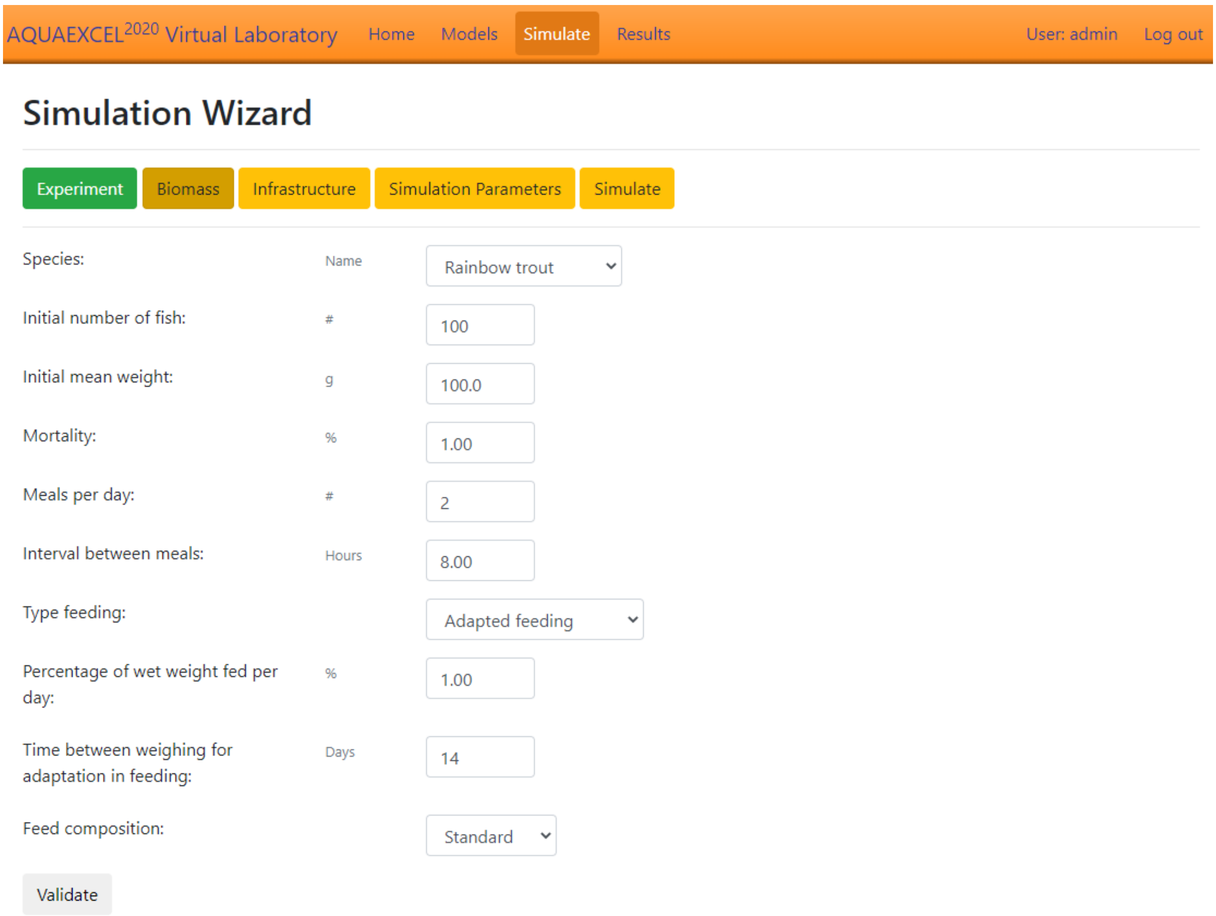


Figure 26 Simulation Wizard Biomass Configuration

To set up your infrastructure you will first need to select either a presimulated tank or a presimulated netpen location, you can also choose to configure your own tank but this will result in flow data not being available.

If you choose a presimulated or a custom tank (see Figure 27), the watertreatment model will be coupled to the growth model, and you need to specify the parameters of the watertreatment system. A presimulated tank will set some parameters automatically and remove them from the user interface, since changing them would result in a custom tank. The parameters that needs to be set are water quality conditions like: temperature, pH, Ammonia, nitrate, solids and total inorganic carbon concentration in the renewal water. System characteristics like: recirculation water flow rate, system water exchange flow rate, total biofilter volume, filling percentage of biofileter, specific surface area of biofilter media, removal efficiency of the drum filter, and the acid-base equilibria carbonate system (K0, K1, K2). Some initial model values also needs to be set in order to initialize the systems, these are: oxygen, ammonia/nitrogen, solids, and total inorganic carbon concentration in the inlet to the fish tank.

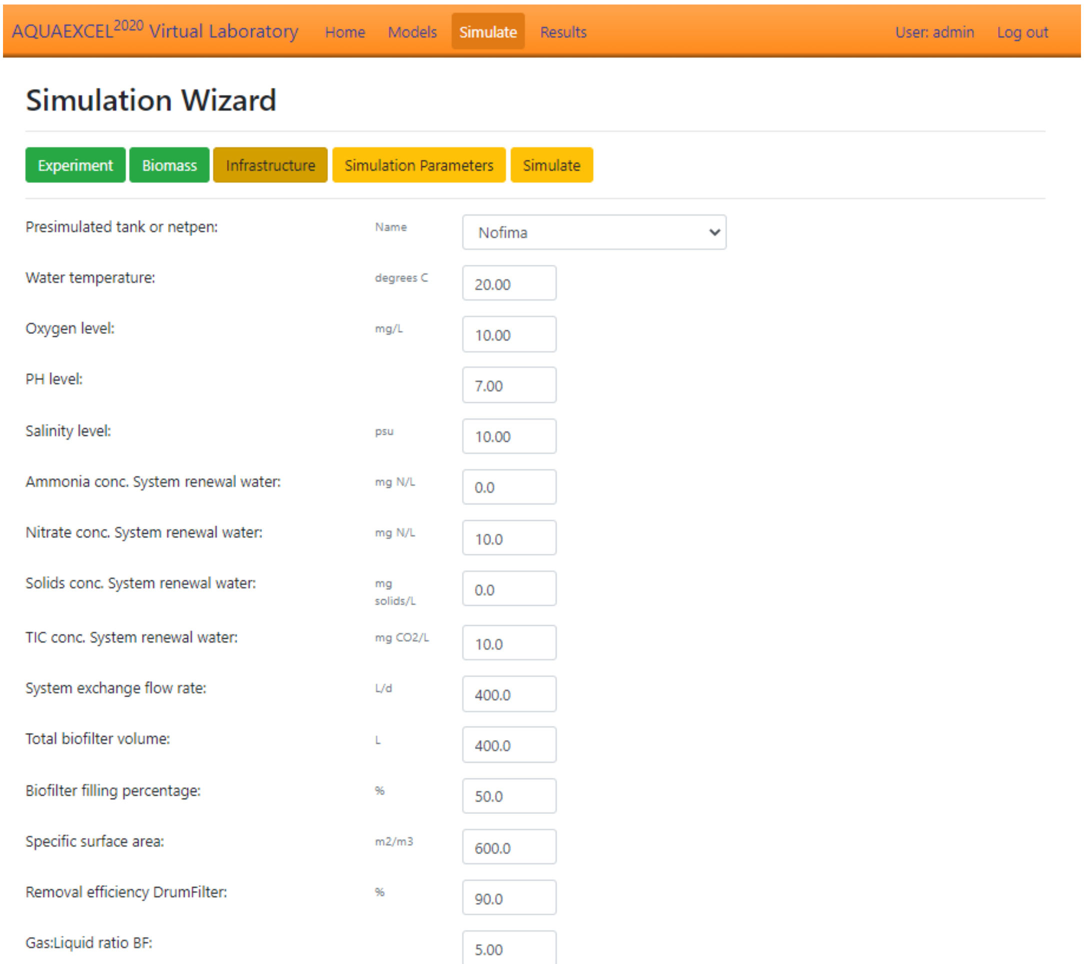


Figure 27 Configure infrastructure with presimulated tank

If you choose a netpen location (see Figure 28), the watertreatment model will be decoupled from the simulation system and far less input parameters will need to be set. You can choose to enable tidal forces on the current at the location, and you will only need to specify the water quality conditions like temperature, oxygen, pH, and salinity level.

Once you are satisfied with your infrastructure, click the validate button and if all values validates it will take you to the "Simulation parameters" page.

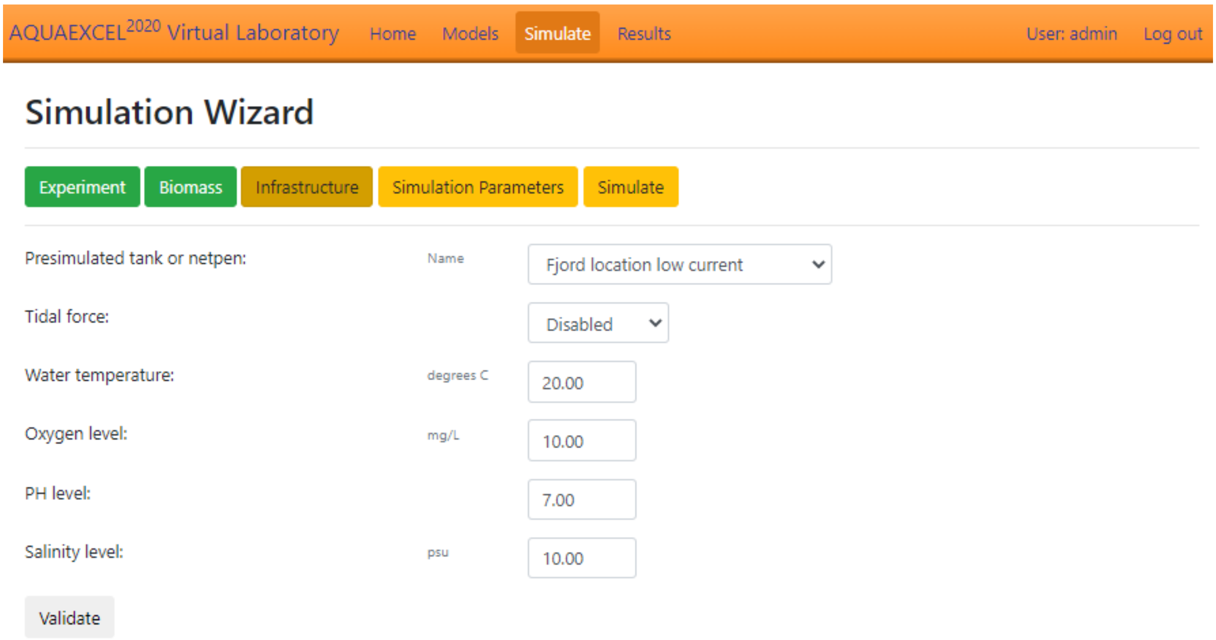


Figure 28 Configure infrastructure with presimulated netcage location

It is possible to go through the experiment wizard in any order by pushing the five different buttons. Figure 29 demonstrates this when we have validated the parameters for the biomass and skipped ahead to the Simulate button. Here we get a status of what has been validated. The Virtual Laboratory will not let us simulate until all parameters are set and validated, at which point the Simulate button will activate and the user may start the simulation.

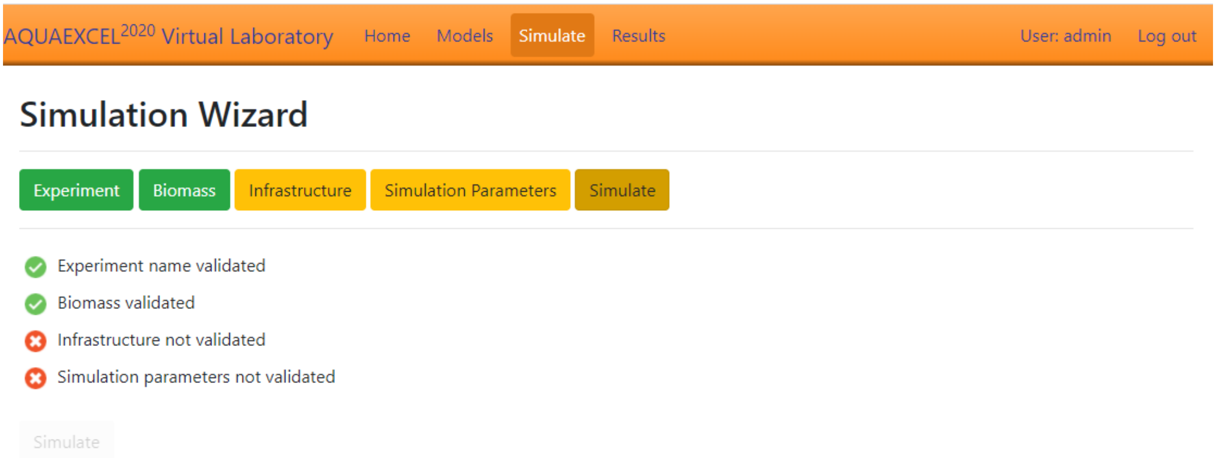


Figure 29 Simulation Wizard Simulation Start

Once the simulation finishes the experiment results will be available from the Results page. Figure 30 shows the listing of different experiments that have been successfully simulated.

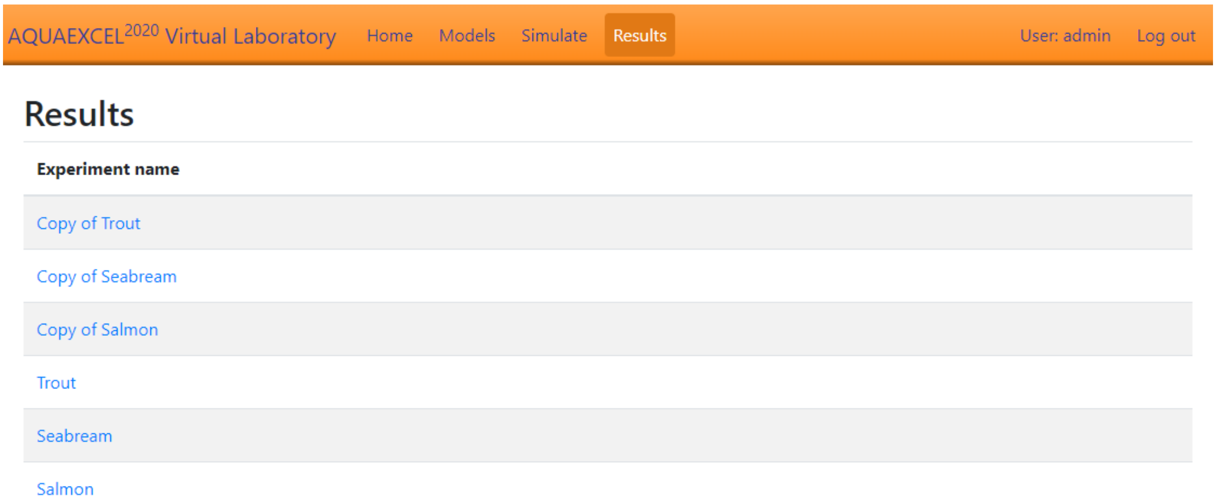


Figure 30 Results Main Page

The user may browse data series from the subpages of the individual experiments. Figure 31 illustrates four graphs relating to the growth model for the chosen experiment. Figure 32 illustrates two graphs relating to the watertreatment model and a graph relating to the flow model. Note that the oscillations in the watertreatment graph corresponds to the feeding graph in the growth model.

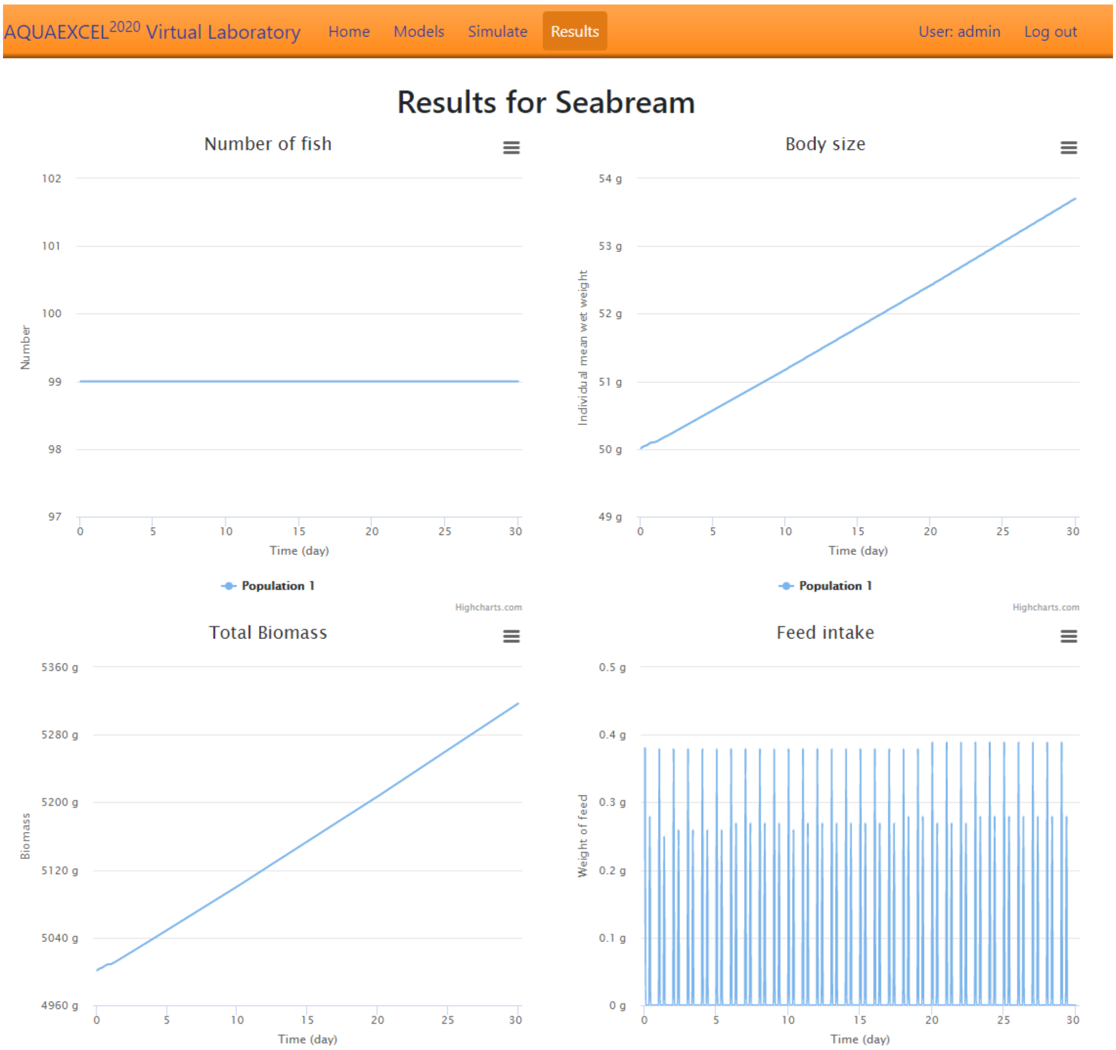


Figure 31 Results Page 1

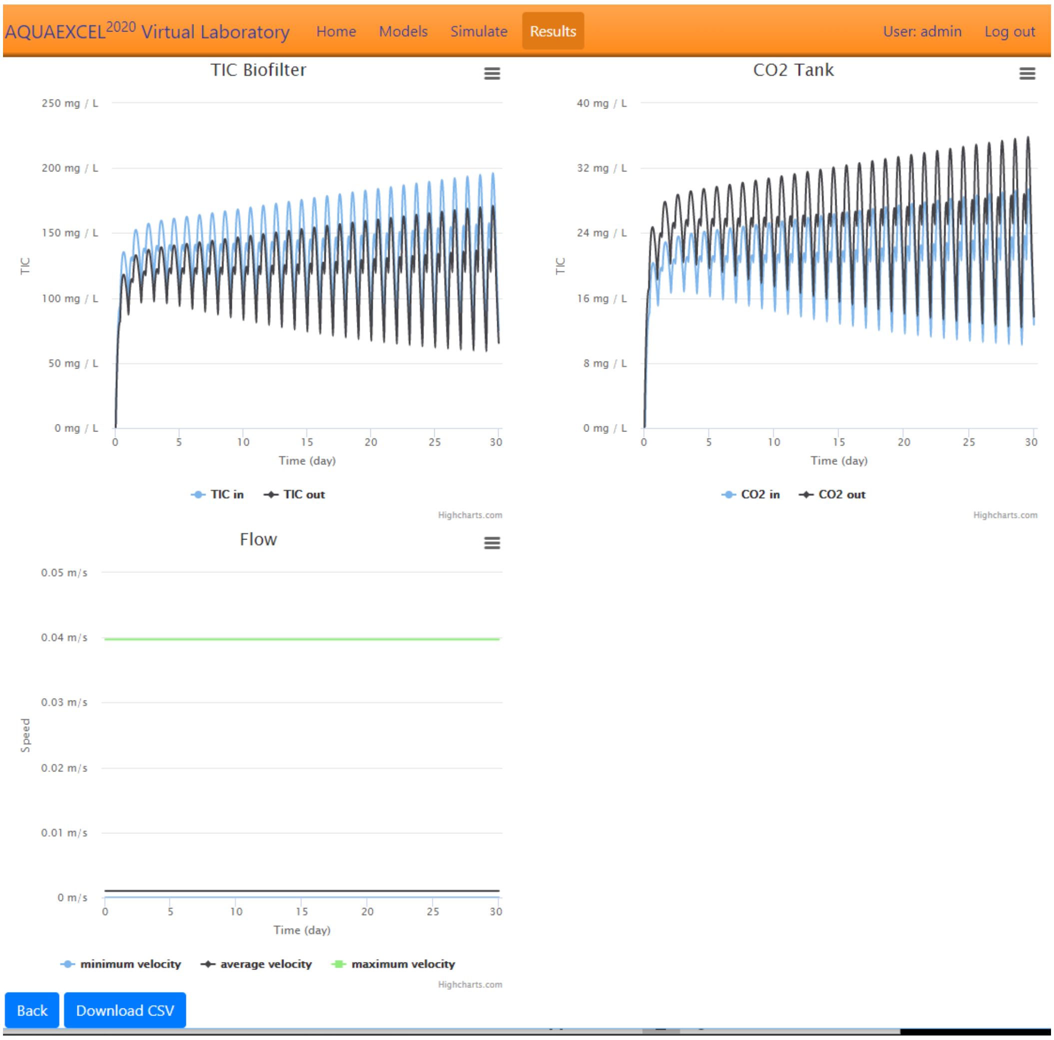


Figure 32 Results page 2

Several key indicators for an experiment have been implemented in the results page, for an overview see Table 1.

Table 1 Results graphs

Model	Description	Parameter	Units
Growth model	Growth	Number of fish	#
		Body size	g fish^{-1}
		Total biomass	g
		Feed intake per fish	g h^{-1}
		Feed conversion ratio	-
	Waste production	Faecal dry matter	g h^{-1}
		Faecal loss-N	g N h^{-1}
		Non faecal loss-N (TAN)	g N h^{-1}
	Gaseous exchange	Oxygen consumption	mg kg^{-1} h^{-1}
		Carbon dioxide production	mg kg^{-1} h^{-1}
Water treatment model	Oxygen	Oxygen fish tank	mg L^{-1}
	Ammonia and Nitrate	Ammonia fish tank in and out	mg L^{-1}
		Ammonia biofilter in and out	mg L^{-1}

		Nitrate concentration system	mg L^{-1}
	Solids	Solids concentration fish tank in and out	mg L^{-1}
	Carbon dioxide	Carbon dioxide concentration fish tank in and out	mg L^{-1}
	Total inorganic carbon concentration (TIC)	TIC fish tank in and out	mg L^{-1}
		TIC biofilter in and out	mg L^{-1}
Flow model	Flow	Minimum velocity	m s^{-1}
		Average velocity	m s^{-1}
		Maximum velocity	m s^{-1}

6. Conclusion

The second version of the Virtual Laboratory has been implemented and we have successfully been able to demonstrate co-simulation of models developed in the sub-workpackages using the architectural principles laid out in Bjørnson et al. (2016).

The web-interface for setting up and executing virtual experiments is available for the public at https://ae2020virtuallab.sintef.no/. The major constraint of the software is that regular users are restricted to co-simulate models that have been pre-selected. However, more advanced simulations are possible for advanced users who know how to operate the underlying framework.

References

Abbink, W. et al. (2018) D5.3 *First prototype model on water quality and water temperature for experimental facilities*, AQUAEXCEL2020 Report.
Abbink, W. et al. (2020) D5.7 *Final model on water quality and water temperature for experimental facilities*, AQUAEXCEL2020 Report.
Alver, M. O. et al. (2018). *D5.4 First prototype flow field model established*, AQUAEXCEL2020 Report.
Alver, M. O. (2020). *D5.8 Final flow field model after testing period*, AQUAEXCEL2020 Report.
Bjørnson, F. O. et al. (2016). *D5.1 Model development guidelines*, AQUAEXCEL2020 Report.
Bjørnson, F. O. et al. (2018). *D5.5 Virtual laboratory version 1*, AQUAEXCEL2020 Report.
Blochwitz, T., *Tutorial: Functional Mockup Interface 2.0 and HiL Applications*, presentation from the 10th International Modelica Conference 2014 available at https://www.fmi-standard.org/literature
Lika, D. et al. (2018) *D5.2 First prototype model on AquaFishDEB*, AQUAEXCEL2020 Report
Lika, K. et al. (2020) *D5.6 Final model for growth, feed consumption and waste production simulation*, AQUAEXCEL2020 Report

Glossary

AQUAEXCEL2020: AQUAculture Infrastructures for EXCELlence in European Fish Research towards 2020

FMI: Functional Mock-up Interface
FMU: Functional Mock-up Unit
CFD: Computational Fluid Dynamics
HTML: Hyper Text Markup Language
CSS: Cascading Style Sheet
JS: Java Script
SVG: Scalable Vector Graphics

Definitions

Domain: Particular area of activity or interest
Software framework: Reusable software environment that facilitates development of software applications, products and solutions
System architecture: Conceptual model that defines the structure and behaviour of a system
Virtual laboratory: An interactive environment for creating and conducting simulated experiments
3R: Guiding principles for more ethical use of animals in testing (Replacement, Reduction, Refinement)

Document information

EU Project N°	652831	**Acronym**	AQUAEXCEL2020
Full Title	AQUAculture Infrastructures for EXCELlence in European Fish Research towards 2020		
Project website	www.aquaexcel.eu		

Deliverable	**N°**	D5.9	**Title**	Virtual laboratory version 2
Work Package	**N°**	5	**Title**	Virtual laboratories and modelling tools for designing experiments in aquaculture facilities

Date of delivery	**Contractual**	30/07/2018 (Month 57)	**Actual**	22/09/2018 (Month 59)
Dissemination level	X	**PU Public, fully open, e.g. web**		
		CO Confidential, restricted under conditions set out in Model Grant Agreement		
		CI Classified, information as referred to in Commission Decision 2001/844/EC.		

Authors (Partner)				
Responsible Author	**Name**	Finn Olav Bjørnson	**Email**	**Finn.o.bjornson@sintef.no**

Version log			
Issue Date	**Revision N°**	**Author**	**Change**
31/08/2020	0	Finn Olav Bjørnson	First version
01.09.2020	1	Eleni Kelasidi	Revision after WP leader inputs
8.9.2020	2	Petr Císař	Revision after review by signed reviewer and coordinator
22.09.2020	3	Finn Olav Bjørnson	Revision after inputs by reviewers