

User Guide for the Diffusion Solver

Rafał Bereś, Yue Chen, Jurgen de Groot, Boyan Kolev, Bernd Rijn

January 31, 2023

1 Introduction

In this user guide the diffusion solver is explained on how it is structured and how to adapt the code for other implementation. This document explains the different files

2 The main structure

The solver is defined into several files and function for a better overview. In this section the different files and functions are explained.

2.1 `SWQ_Diffusion_solver.jl`

`SWQ_Diffusion_solver.jl` is the main file of this program. In this file you can execute the code by choosing different parameters in the `Param`-function and running the `run`-function. One could also choose to execute only certain parts of the code by first calling the `setup`-function which contains the setup parameters and then either the `Spinup`-function or the `SWES_Diff_dt`-function which executes the solving part of the function. When the same problem is repeatedly done, one could run the `Spinup` function once and use the output as the initial condition for the `SWES_Diff_dt`-function for all other cases.

2.2 `SWQ_Parameters.jl`

This file contains all the parameters used in the program. One could adapt the variables in this file or change the variables in the main file. All of the variables correspond with the definition used in the report. Some control parameters are added for turning off certain function. By putting these parameters to "false", the function (or parts of the function) will be excluded from the Residual and Jacobian. There should always be a function depended on the test functions in the integrals over the domain and boundary so when all of the control parameters are set to "false" the code will give an error message.

2.3 `SWQ_DIFFUSION_FUNCTIONS.jl`

This file contains all the functions used in the Residual and Jacobian. If desired, one could change the functions and access the parameters with `@unpack`. The forcing function `forcfunc` is also in this file.

2.4 `SWQ_Initial_Solution.jl`

This file contains the initial solution together with the function to define h . This could be changed by either changing the parameters in the `Param`-function or defining a function inside the scope. The same holds for h . A example on how to do this is also in the file and could be activated by removing the comments.

2.5 `SWQ_Make_Model.jl`

This files makes the model for the implementation. This code could be changed in a way to implement a GMSH-model.

2.6 SWQ_Setup.jl

The setup file makes the model and makes the initial conditions.

2.7 SWQ_Solver.jl

The solver file defines the Residual, Jacobian and the Solver. One could change the functions in here or define a different type of solver.

2.8 SWQ_Write_Output.jl

This file contains the solving of the function and the saving. The file will make the direction, save it to a PVD with different VTU files. It could also be saved at certain points to a CSV file. This could be adopted in the Parameters.