LIQUID:
2-D Fluid Dynamics Simulator

Software Requirements Specification

V1.0

October 06, 2015

Tech Geeks
Ayman Almusalam
Michael Chase Bonifant
Sai Krishna Kalakonda
Srikanth Reddy Bitla
Nagasindhu Kannekanti

Prepared for
Principles of Software Engineering
Fall 2015

LIQUID:
2-D Fluid Dynamics Simulator

Revision History

Date	Description	Author	Comments
9/21/2015	Maintainability, Portability and Design Constraints	Ayman Almusalam	
9/21/2015	Functional Requirements	Michael Bonifant	
9/22/2015	Scope and Assumptions	Sai Krishna Kalakonda	
9/22/2015	Requirement Evaluation	Srikanth Reddy Bitla	
9/22/2015	Introdroduction and General Description	Nagasindhu Kannekanti	

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date

LIQUID:
2-D Fluid Dynamics Simulator

Table of Contents

LIQUID:
2-D Fluid Dynamics Simulator

1. Introduction

1.1 Purpose

The purpose of this document is to present the specifications and requirements of the 2-D Fluid Dynamics Simulator system. It will explain the purpose and features of the application, the interface of the application, what the application will do. This document is intended for both the stakeholders and the developers of the system.

1.2 Scope

The single software product to be produced is the 2D Fluid Dynamics Simulator system. The system will provide simulations for single fluids in a configurable simulation/virtual chamber. It will show in 2D the flow density of the fluid being simulated and allow the user to specify basic environmental factors about the simulated environment like obstructions in the chamber and the viscosity of the fluid being simulated.

This tool will be beneficial as a learning tool for introducing students to fluid dynamics. It will not be useful in simulating large scale fluid systems (for instance, the flow pattern of run off from rain over a textured surface, the human circulatory system, or a city sewer system).

Once the Testing application has been deployed and is fully operational, any support or maintenance is out of the scope for this project.

1.3 Definitions, Acronyms, and Abbreviations

Lattice Boltzmann Methods (LBM): a class of computational fluid dynamics methods used in fluid simulation from which the Navier-Stokes equations can be derived.
Chamber: when used in this document chamber refers to the simulated container that the fluid dynamics are being simulated inside of.
System: when used in this document refers to the deliverable end product, the 2D Fluid Dynamics Simulator.

1.4 References

https://www.ibiblio.org/e-notes/webgl/gpu/fluid.htm
http://www.kevinbeason.com/scs/fluid/
http://http.developer.nvidia.com/GPUGems/gpugems_ch38.html
http://physics.weber.edu/schroeder/fluids/
http://www.cims.nyu.edu/~billbao/report930.pdf
https://en.wikipedia.org/wiki/Lattice_Boltzmann_methods
"Fluid dynamics in Group T-3 Los Alamos National Laboratory:(LA-UR-03-3852)".

LIQUID:

2-D Fluid Dynamics Simulator

1.5 Overview

The following sections of this document detail the system in a mile -high description such as might relate to a sales pitch and explaining the system to a layman (section 2), the fine grained system requirements from which developers should be able to build their own version of the system (section 3), detailed descriptions of analysis models used elsewhere referentially (section 4) and additional appendices.

2. General Description

2.1 Product Perspective

This system will serve as a self-contained desktop application for simulating 2-D fluid dynamics. It should be a simple, intuitive tool that allows users to configure a chamber filling or filled with fluid, vary environmental factors (e.g. obstructions in the chamber, temperature, viscosity), then apply a force vector to disturb the fluid and show a simulation of that force's effect on the fluid.

2.2 Product Functions

The system allows the user to create a simulation chamber for testing what happens when a force vector is applied to a fluid filling that chamber. As such the system allows users to select a fluid and a force vector. Additionally they can set environmental variables such as the temperature among others (see section 3.2.2 for a full listing of configurable variables). The user can also create within the chamber any number of physical obstructions the fluid can potentially rebound against and flow around.

The user can then set monitors at various points in the chamber. Once the chamber is fully configured the user can launch the simulation and run it until they choose to terminate the simulation. The simulation runs showing a visual display of the created chamber and also provides a textual print out at the state of the simulation at all break points.

From the textual print out the will be able to reload the simulation and replay the simulation.

2.3 User Characteristics

Users for this system include students in high school and beyond learning the basics of fluid dynamics and their teachers.

2.4 General Constraints

The main constraints are the system must be a desktop application, and it must run on either Linux or Windows. Beyond that the simulation must be based on the LBM.

2.5 Assumptions and Dependencies

- It's assumed the system will be run on hardware supporting the operating system the designer chooses to develop the system for stand alone.
- Custom User Interface design or theme
- With a detailed test plan, it is assumed that the client will be able to sufficiently maintain the system without requiring help from external parties.
- Development, installation and testing of the system are dependent on server space being made available, set up and provided by Information Services in conjunction with the client.
- The project team (as students) will be able to access all software required for development of the system. Where possible, open source or freeware software will be used.
- Initial testing will be conducted by the client through a prototype.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interface GUI shall be user friendly and provide an animation of the simulation being run.

3.1.2 Hardware Interfaces

The system has no hardware interface requirements.

3.1.3 Software Interfaces

The system has no software interface requirements.

3.1.4 Communications Interfaces

The system has no communications interfaces, it works as a self-contained application.

3.2 Functional Requirements

3.2.1 2-D Fluid Simulation

3.2.1.1: Introduction

The system shall visualize flow density dynamically.

3.2.1.2: Inputs

The system takes as input a single user selected fluid (see 3.2.3), physical parameters (see 3.2.2), break points (see 3.2.6), and a spatial configuration of the simulation (see 3.2.4).

3.2.1.3: Processing

The system must use the given inputs to simulate the flow density via the LBM. The fluid will never overflow the simulation chamber. Visualization will always be from a top down/floor plan point of view. Gravity if relevant to the LBM is constant and considered the average measurement of gravity near the surface of the earth.

3.2.1.4: Outputs

Animated and textual feedback pertaining to the flow density as time progresses from the initial state specified by the input parameters. This feedback is constantly updated step by step until the user chooses to stop execution.

3.2.1.5: Error Handling

The core functionality running the LBM should not fail. If it does then the user should be able to generate an error report to send to the support team that will detail the initial launch of the simulation and the user's system so support can determine why LBM itself failed on their system.

3.2.2 Physical Parameters

3.2.2.1: Introduction

The system shall allow the user control in varying the viscosity, temperature, initial steady state flow speed, and perturbation force of the simulation.

3.2.2.2: Inputs:

The viscosity, temperature (from a range of -100C to 100C), flow speed, and perturbation force vector. There is no specification as to how these are to be obtained

3.2.3.3: Process:

There is no specification to how this data need be processed beyond it must be utilized in the calculations of the fluid flow density with the LBM.

3.2.2.4: Output:

There is no specification as to the output beyond that they must be used in calculating the flow density of the fluid being simulated, and should be somehow incorporated into the textual feedback of the simulation so that the simulation could take that text and reproduce the simulation exactly.

3.2.2.5: Error Handling

User cannot be allowed to set impossible fluid simulations, like trying to simulate water at 1 atm and 0C or 100C where water is no longer a fluid and instead either solid or gaseous.

3.2.3 Fluid Selections

3.2.3.1: Introduction

The system shall provide simulations for water and glycerin.

3.2.3.2: Inputs

The user's selection of water or glycerin.

3.2.3.3: Process
The system must track the fluid's change in flow density using LBM.

3.2.3.4: Outputs
The user should get visual feedback of what's happening in the simulation chamber and also textual output.

3.2.3.5: Error Handling
There is no specified error handling.

3.2.4 Spatial Configuration

3.2.4.1: Introduction
User shall be able to define rectangular boxes and elliptic cylinders as obstructions to the fluid's flow in the simulation chamber. the chamber itself will be rectangular visual from above, though it may have rounded sides, this curving is also configurable.

3.2.4.2: Inputs
The user will be able to select the objects to add to the simulation, via the GUI. (The most user friendly way is probably by dragging and dropping them into the simulation from a selection pool of the predefined shapes and resizing them inside the display like when inserting boxes and ellipses in MS PowerPoint as an example.)

3.2.4.3: Process
Beyond the fact that the simulation must account for their existence and adjust the flow of the fluid accordingly with LBM the simulation may ignore the edge cases where the fluid hits sharp well defined cusps.
Additionally all obstructions span from the floor of the chamber to the roof and considered immutable (i.e. the fluid cannot create such a force as to break the obstructions or move them).
Lastly the chamber can be on a scale ranging from centimeters in size to meters.

3.2.4.4: Output
There is no specification as to how these obstructions or chamber shape are to be shown as output, beyond that there must be some animated feedback in the GUI and human readable textual output.

3.2.4.5: Error Handling
Objects with cusps cannot be oriented so the fluid hits the cusp head on.

3.2.5 On Demand Feedback

3.2.5.1: Introduction
The system shall be able to provide on-demand information about flow values.

3.2.5.2: Inputs
The user can select any point in the simulation chamber and immediately get feedback as to the flow values of that location in the simulation immediately.

3.2.5.3: Process
There is no specification to the process this need be implemented by, beyond the returned values must agree with the LBM.

3.2.5.4: Output
The flow density at all points needs to be visualized in animation and text already, requesting this feedback is merely highlighting the textual output for the selected region.

3.2.5.5: Error handling
There is no error handling specified.

3.2.6 Monitoring Points

3.2.6.1: Introduction
The user shall be able to select monitoring points in the simulation (flow meters) to show information during execution without affecting the flow of the fluid.

3.2.6.2: Inputs
There is no specification as to how there flow meters be obtained, beyond the user must provide them.

3.2.6.3: Processing
There is no specification as to how these flow meters need to be implemented beyond that their output must agree with LBM.

3.2.6.4: Output
Textual feedback of the flow density at the specified locations in the simulation.

3.2.6.5: Error Handling
There is no specification for error handling.

3.2.7 Clear Start and End of Simulation

3.2.7.1: Introduction
The system shall indicate clearly when a simulation begins and terminates.

3.2.7.2: Inputs
The user selects when to start the simulation and when to terminate it. There is no specification as to how this information is to be selected, beyond once it starts the user can terminate the program at any moment.

3.2.7.3: Process
There is no specification as to how this is to be implemented

3.2.7.4: Output
Clear animated and textual feedback that the simulation has started and terminated.

3.2.7.5: Error handling
There is no specification for error handling.

3.2.8 Textual Logging

3.2.8.1: Introduction
The system shall provide a detailed log of the execution of the simulation so that the user can go back through a text file and trace the execution of the program.

3.2.8.2: Inputs
There is no specification as to inputs, (but presumably a name for each simulation to attribute to the generated text file is required).

3.2.8.3: Process
There is no specification as to how the text is to be generated or worded, beyond it must provide feedback for all marked flow meters, and the system as a whole.

3.2.8.4: Output
A text file containing a trace of the execution of the simulation.

3.2.8.5: Error Handling
There is no specification for error handling.

3.2.9 Instant Replay

3.2.9.1: Introduction
The system shall allow the user to replay any simulation by reloading the simulation from its generated textual log file.

3.2.9.2: Inputs
The selected log file, there is no specification as to how this file be obtained. (Unique naming system would work simply by using the time stamp the simulation is run at.)

3.2.9.3: Process
The system must load the log file and execute reproducing the original simulation exactly.

3.2.9.4: Output
Animated and textual feedback about the simulation, the simulation should be identical to the

original.

3.2.9.5: Error Handling
There is no specification for error handling.

3.2.10: Generic Error Handling
The Results of the simulation can be trusted and depends on the degree of uncertainty and on various errors.
Uncertainty is something which is defined as potential deficiency.
Errors are again classified as

1. Known or Acknowledged Errors like Round-off errors due to finite precision of computer arithmetic.
2. Unknown and Unacknowledged Errors like bugs in the code and any logical mistakes
3. Local Errors are the part of Global errors and may affect throughout the simulation
4. Global Errors are collection of many local errors

Error handling should include

- Log information about the system state, running process and errors which are encountered
- Should able to detect failed services
- Should provide full information about failures and errors that includes time of failure, origin, severity and other diagnostic information

Log shall be saved in some independent data storage

3.3 Use Cases

3.3.1 Filling a Chamber

The system will allow the user to simulate a fluid filling the chamber from empty until it is full. In this case the flow rate of fluid into the chamber must be constant.

3.3.2 Perturbing a Chamber

The system will allow the user to simulate a single force perturbing the fluid in the chamber. This force cannot be applied from a corner of the chamber, only from one of its sides.

3.3.3 Draining a Chamber

The system will allow the user to an exit way for the chamber.

3.4 Classes / Objects

There are no class/object requirements.

3.5 Non-Functional Requirements

3.5.1 Performance

3.5.1.1: The system overall shall produce results accurate to the 4^{th} decimal place.
3.5.1.2: The system shall produce results in metric units.

3.5.2 Reliability

There are no reliability specifications.

3.5.3 Availability

The system is to be available as a self-contained desktop application.

3.5.4 Security

There are no security specifications.

3.5.5 Maintainability

3.5.5.1: The system shall be designed so that the GUI and simulation engine can be fully decoupled and swapped out with other GUIs and engines.
3.5.5.2: Distributions of the software shall include the source code.
3.5.5.3: Distributions of the software shall include compilation instructions
3.5.5.4: Distributions of the software shall include a list of dependencies.
3.5.5.5: Distributions of the software shall include either an installer or installation script.
3.5.5.6: Distributions of the software shall include all generated document artifacts during the engineering life cycle.
3.5.5.7: Distributions of the software shall include an installation guide
3.5.5.8: Distributions of the software shall include a user guide with screen shots.
3.5.5.9: Distributions of the software shall include a programmer's guide.
3.5.5.10: All generated documents shall be typed and electronically generated.

3.5.6 Portability

3.5.6.1: The system must work on either a Windows or Linux operating system, with no expectation that it be portable from it original target operating system.

3.6 Inverse Requirements

3.6.1: The application does not simulate multiple fluids interacting, diffusing or mixing in any way, only one fluid is simulated at a time.
3.6.2: The application only simulates a single perturbing force from a wall (not corners/cusps), if a fluid is filling the chamber, that fluid is already providing the perturbing force, so a second is not specifiable. This is also true if the chamber is draining.

3.7 Design Constraints

3.7.1: The system shall simulate fluid dynamics using LBM.
3.7.2: The system must be an installed desktop application for either a Linux or Windows environment.
3.7.3: The system must be released under GNU GPL3 licensing.
3.7.4: The system may optionally include ONE third party component with the customer's approval of the component.

3.8 Logical Database Requirements

There is no requirement that a database be used.

3.9 Other Requirements

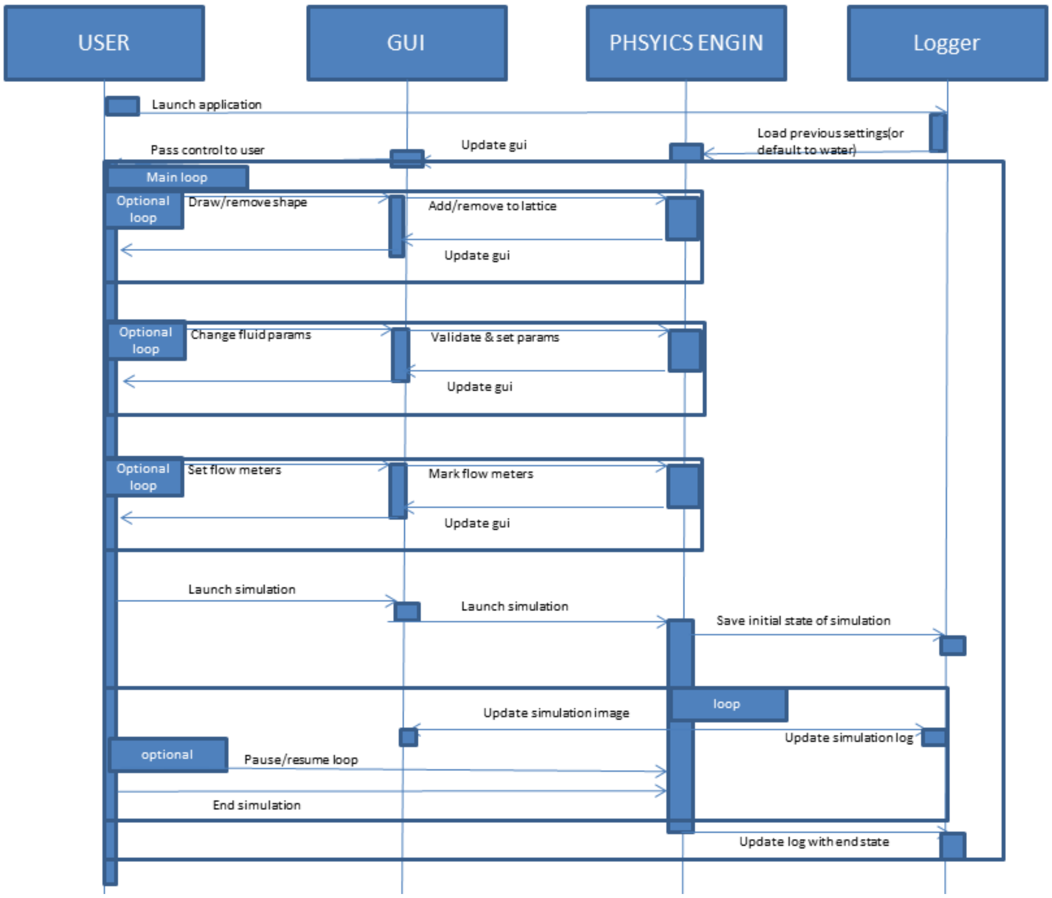
There are no other known requirements.

4. Analysis Models

4.1 Sequence Diagrams

Basic flow of running a simulation. The user launches the application which defaults to the user's last settings (or a basic water simulation if no prior settings). The user can optionally add shapes (rectangles and ellipses) to the simulation as solid barriers for the fluid. The user can set the fluid parameters (change the fluid, temp/viscosity, etc). The user can optionally set flow meters. (These three tasks can occur in any order (note: the user can't put a flow meter over a solid barrier or a solid barrier over a flow meter).)

LIQUID:
2-D Fluid Dynamics Simulator

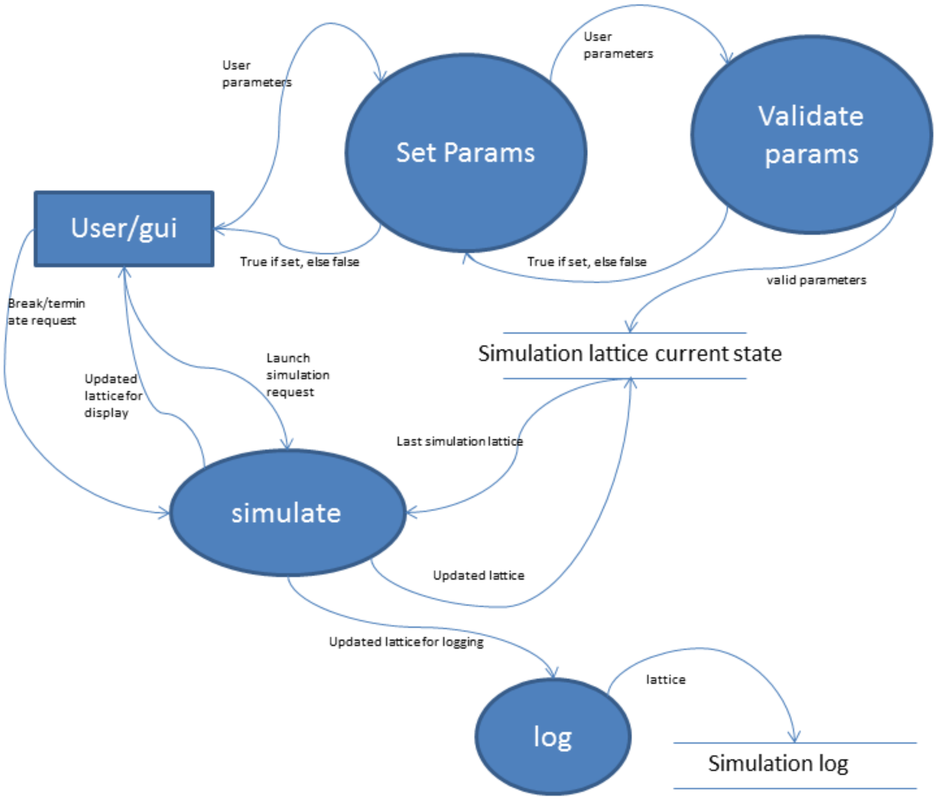


The user then launches the simulation which they can break/pause at any moment. The Simulation progresses in increments logging the simulation state at each step and outputting the simulation display and output of flow meters.

At any moment the user can end the simulation and start over, defaulting to the set up as they last configured it or to the default water simulation.

4.2 Data Flow Diagrams (DFD)

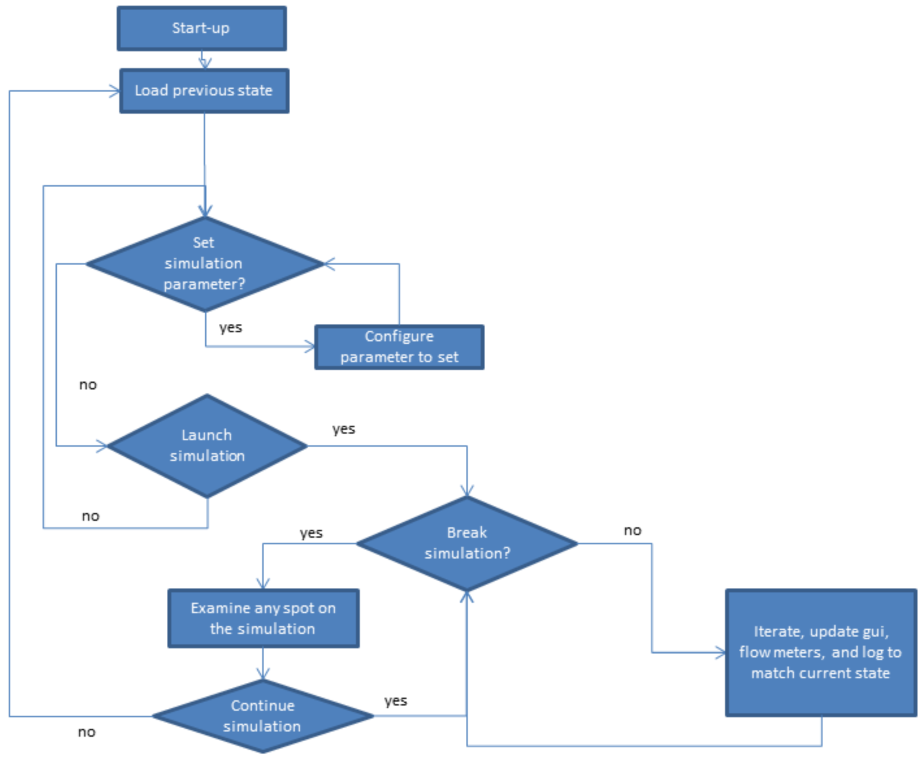
The main source of input/outputs is the user/gui, through which the user attempts to set parameters for the simulation (if valid (ie no trying to simulate ice)) stored in the simulation lattice current state).



They can additionally launch the simulation which applies one iteration of the lattice Boltzmann methods to the current lattice, logs the state, updates the current state and GUI, then iterates again and loops until a request to pause or terminate (until terminated no further params can be set).

4.3 Other Flow Diagrams

User starts, system loads their previous settings, the user can then loop through setting parameters (changing the simulation chamber size, adjusting the fluid, adding flow meters, etc) until they're satisfied with the set up. Then they can launch the simulation, and at each iteration of it have a chance to pause it. While paused they have an option to terminate the simulation which then sets the simulation back to how it was when it was right before it was run.
At each iteration of the simulation the visual display, flow meter outputs, and log are updated.



5. Change Management Process

As we are heading forward with the project with agile methodology, there is always a scope for new changes or Change Requests. Taking this into consideration, we would be able to modify or edit the SRS document as per the requirement and will be writing the document in LaTeX in TEXMaker. Also we are adopting the Github for our revisions management. Asana (Project management tool) is already in place to track all of our tasks. All the revisions are published on Github and shared among the team member. However final version need to be approved by client.

A. Appendices

There are presently no appendices.