

# Vortexy fluid dynamics simulator

Roni Koitermaa (Roninkoi)  
roninkoi@iki.fi

August 7, 2019



# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	Navier-Stokes equations . . . . .	3
2.2	Finite volume method . . . . .	3
2.3	Discretization . . . . .	3
2.4	Jacobi method . . . . .	3
<b>3</b>	<b>Implementation</b>	<b>3</b>
3.1	Simulation mesh . . . . .	3
3.2	Solver . . . . .	3
3.3	Parallelization . . . . .	3
3.4	Rendering . . . . .	3
<b>4</b>	<b>Technical documentation</b>	<b>3</b>
4.1	Compilation . . . . .	3
4.2	Code style . . . . .	3
4.3	Configuration . . . . .	3
4.4	Examples . . . . .	3
	<b>References</b>	<b>3</b>

# 1 Introduction

**Vortexy** (finnish: Vorteksi) is a computational fluid dynamics (CFD) simulation package. It is written in C and uses OpenCL to utilize graphics processing units (GPU). Included is also a renderer that uses OpenGL to visualize results.

The simulator takes a configuration file as input that contains paths to the simulation mesh, velocity fields and boundary conditions in addition to other settings. The finite volume method is then used to calculate the time evolution of the system and this is visualized on the screen. The state of the system is periodically written to an output file specified in the config.

## 2 Background

### 2.1 Navier-Stokes equations

$$\frac{\partial u}{\partial t} + (u \cdot \nabla)u = -\frac{1}{\rho}\nabla p + \nu\nabla^2 u + g \quad (1)$$

### 2.2 Finite volume method

### 2.3 Discretization

### 2.4 Jacobi method

## 3 Implementation

### 3.1 Simulation mesh

### 3.2 Solver

### 3.3 Parallelization

### 3.4 Rendering

## 4 Technical documentation

### 4.1 Compilation

### 4.2 Code style

### 4.3 Configuration

### 4.4 Examples

## References