

Assignment 3, Report

Maximilian Mayerl, Alexander Schlögl, Benedikt Wimmer

January 9, 2018

1 Implementation

1.1 Task 1 - Semi-Lagrangian Advection

For solving the advection we need to calculate the quantities for the current step, for each node. This is done as follows:

1. Calculate the old coordinate by subtracting $v \cdot \Delta t$
2. Determine the four surrounding nodes in the grid
3. Interpolate their quantity values

The new values are then stored in the `*tempField` variable.

1.2 Task 2 - Gauss-Seidel Solver

We implement the Gauss-Seidel method exactly as described on the lecture slides. It is pretty simple since all calculations are done on the same pressure field, using old values as initialization and combining the four surrounding values to a new one. We calculate the residual after each step and stop iterating after it is below the accuracy threshold. We calculate the residual in a second pass because we were unsure whether the incomplete computation of the pressure field matters too much to do it in one pass. The only problem would be the solver ending prematurely with the actual field not having the desired accuracy. In practice the decreased computation time combined with the (possibly) required higher accuracy (a lower accuracy threshold) is still better than doing two passes. For this exercise it should be fine as it is, however.

1.3 Task 3 - Velocity Update

We implement the component-wise velocity update exactly as described on the lecture slides. The new velocities are calculated from the old velocities and the new pressure field like:

$$v_x^{n+1} = v_x^n - \Delta t \left(\frac{1}{h} (p^{n+1}(i, j) - p^{n+1}(i-1, j)) \right)$$

$$v_y^{n+1} = v_y^n - \Delta t \left(\frac{1}{h} (p^{n+1}(i, j) - p^{n+1}(i, j-1)) \right)$$