

# Real-time Control and Stopping of Fluids

Károly Zsolnai and László Szirmay-Kalos

Budapest University of Technology and Economics, Hungary

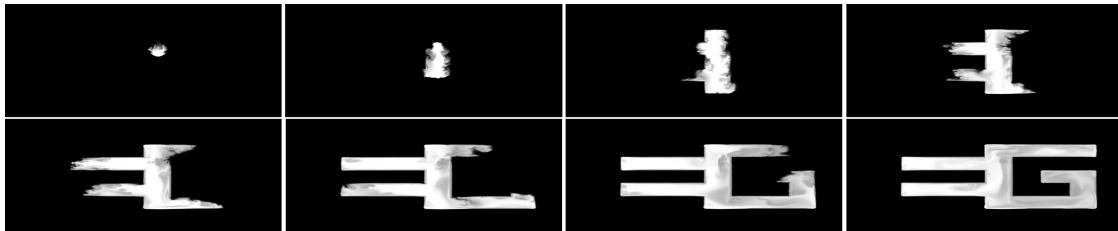


Figure 1: Animating the fluid into the EG logo with the proposed method.

## Abstract

In this paper we address the fluid control problem, where an arbitrary density distribution (a shape of any kind) is given, and forces are exerted to get the fluid to flow into this shape and stop when the target distribution is reached. We present a real-time solution.

## 1. Introduction

Fluid simulation means the mimicking of real fluids by solving the Navier-Stokes equations:

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} = -\frac{1}{\rho} \nabla p + \frac{\mu}{\rho} \nabla^2 \vec{u} + \vec{a}_{\text{ext}}, \quad (1)$$

where  $\vec{u}$  is velocity field,  $\rho$  stands for the density,  $p$  for pressure,  $\mu/\rho$  denotes the kinematic viscosity of the fluid, which is the ratio of dynamic viscosity  $\mu$  and the density, and  $\vec{a}_{\text{ext}}$  is the acceleration caused by external forces. The advection term  $(\vec{u} \cdot \nabla) \vec{u}$  shows up because the fluid element is not followed in the Eulerian viewpoint, but the location in focus is fixed to the lattice points of a static grid. This equation expresses the conservation of momentum. In addition, we should also enforce mass conservation by  $\nabla \cdot \vec{u} = 0$ .

Fluid control, on the other hand, is the determination of parameters in a way that the resulting fluid motion follows a prescribed behavior. We have to find external forces that make the density field converge to a target density. Several solutions exist to solve the fluid control problem, such as Jos Stam's adjoint method [MTPS04], or the control of fluids on triangle meshes [RTWT12]. The algorithm of Shi and Yu adds a long range force field to even out the distribution on macroscopic level, and a short range field to carve out the fine details [SY04]. Short range force field is obtained as minimizing a functional that ensures that density change

is maximal in excess density areas, the flow is divergence free and the alignment of the velocity and the gradient of the density. The computation of the short range forces may take several minutes in each frame.

Fluid stopping is an essential part of fluid control. When the target density is obtained, the converged state must be maintained in a natural way. The most intuitive solution for this is to increase the viscosity, i.e. the friction to dissipate the kinetic energy and to make the fluid stop.

## 2. The new method

In this paper, we present a simple method for real-time fluid control. We aim at a dynamic balance where the objective is satisfied by a constantly moving fluid. Instead of complicated short range force computation, we let the pressure field do the local tuning of the densities. To help the pressure field, when the target density is approximately reached, the total internal force, including both the pressure and the friction, is scaled up.

Our long range force field is similar to [SY04]. If some part of the fluid domain has excess density, meaning that the density at point  $j$ ,  $\rho_j$ , is higher than the target density  $\rho_j^t$  given by the input distribution, the region will transport density by exerting force towards the direction of its neighborhood for those who have lower density than the target.

The exerted force weakens with the square distance. The acceleration due to this force field is

$$\vec{a}_i^L = c \sum_j \frac{[\rho_j - \rho_i^t]^+ \vec{r}_{ij}}{|\vec{r}_{ij}|^3}, \quad (2)$$

where  $c$  is a constant factor to control the magnitude of the control force field,  $\vec{r}_{ij}$  points from grid point  $i$  to  $j$  and superscript  $+$  denotes replacing negative values by zero.

Fluid stopping strategy decides what happens after the target density is reached. The most straightforward solution is to increase the viscosity to a very high value to “freeze” the fluid in convergent subdomains. The results will remain correct, but not very lifelike and generally unconvincing. Here we address the shortcomings of this approach by introducing a *scaling factor*  $s$  not only for the friction but for the total internal force of the fluid, which includes the pressure as well. This idea may sound counterintuitive: why speed up the fluid at regions where it already looks correct? Let us consider an example, where the target distribution can be reached only by going through a narrow choke point. When the fluid starts freezing, it prevents further fluid movement, making it impossible to get density through. This scenario shows up for almost every practical case on closed shapes when the system gets close to the state of convergence.

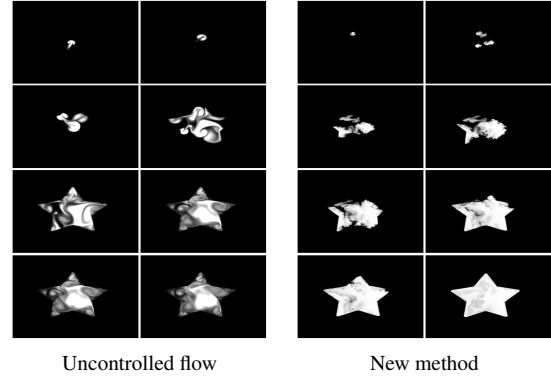
Note that scaling both the friction and the pressure is equivalent to decreasing the density while keeping the dynamic viscosity constant. As the fluid is stable for arbitrary positive density because the energy is dissipated by the friction, the introduced scaling does not endanger the stability of the simulation. Intuitively, classical freezing of the fluid can be associated with the “after you are done, just stop and rest” behavior, as opposed to simultaneously scaling up the pressure and the friction, which would mean more like “after you are done, start helping others”. This behavior will not only allow the fluid to flow through narrow choke points, but effectively transfer density to neighboring regions of poor convergence and preserve fluid movement after the target density is reached.

Putting it all together, the Navier-Stokes equation is modified and the internal force is scaled up by  $s$  where the actual density is close to the target density. The scaled up internal force will maintain some motion even close to the converged state when the control force drops to zero. The modified equation is:

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} = s \left( -\frac{1}{\rho} \nabla p + \frac{\mu}{\rho} \nabla^2 \vec{u} \right) + \vec{a}_{\text{ext}}, \quad (3)$$

$$s = \begin{cases} 1 + \delta, & \text{where } |\rho^t - \rho| < \varepsilon, \\ 1, & \text{elsewhere.} \end{cases}$$

where  $\delta$  is small value to keep the fluid in motion after convergence when the actual density is equal to the target density within error threshold  $\varepsilon$ . This technique is capable of guiding the fluid towards the target distribution in real-time.



**Figure 2:** Fluid simulation without control forces, only boundary conditions are used (left). The example shows that even if the fluid is locked inside the domain of interest, it is highly unlikely that it would suddenly flow into the shape of a star. The proposed method provides good coverage of the target density, and is aware of the regions of poor convergence, which are constantly helped out by nearby regions (right). The same amount of density is used in both cases.

### 3. Results

We compared the new method to the uncontrolled one, where Neumann-type boundary conditions are defined but no control force field is applied. This uncontrolled flow does not have a desirable degree of convergence (left image of Figure 2). The new is unable to handle any amount of density which is not near or inside the target distribution domain. The results using the new method were rendered in real-time and are shown in the right image of Figure 2, which, using the same amount of density, was able to achieve fast convergence.

### 4. Conclusions

In this paper, we proposed a novel algorithm to the fluid control problem, where we scaled up the total internal force in converged regions instead of computing complicated short range force field and freezing the fluid.

### Acknowledgement

This work has been supported by OTKA K-104476.

### References

- [MTPS04] MCNAMARA A., TREUILLE A., POPOVIE Z., STAM J.: Fluid control using the adjoint method. *ACM Trans. Graph.* (2004), 449–456.
- [RTWT12] RAVEENDRAN K., THUEREY N., WOJTAN C., TURK G.: Controlling Fluids using Meshes. *SCA* (2012).
- [SY04] SHI L., YU Y.: Inviscid and incompressible fluid simulation on triangle meshes. *Journ. of Comp. Anim. and Virt. Worlds* 15 (2004), 3–4