

# Real-time control of Newtonian fluids using the Navier-Stokes equations

Károly Zsolnai and László Szirmay-Kalos

Department of Control Engineering and Information Technology, Budapest  
 University of Technology, Hungary  
 zsolnai@iit.bme.hu, szirmay@iit.bme.hu

**Abstract.** In this paper we address the fluid control problem, where, alongside simulating the motion of fluids, an arbitrary density distribution (a shape of any kind) is given, and forces are exerted on the system with the intention that the fluid would sooner or later take this form. Prior research work has shown that the problem is challenging due to multiple reasons: first, every region is tightly bound to its neighborhood, therefore a force that acts upon a point in space will also have effect on nearby regions, making the controlled process strongly coupled. Second, it also a desirable requirement that the controlling external force field should make the fluid flow realistic, even though it is highly improbable that a given volume of water would suddenly flow exactly into a given shape. Utilizing sophisticated mathematical methods from differential fields such as optimization and control theory, current state of the art techniques are able to give visually pleasing results at the cost of 5 to 7 minutes of computation time per frame. We present a novel solution for the fluid control problem with certain restrictions, making it possible to solve it in real-time.

## 1 Introduction

*Fluid simulation* means the mimicking of real fluids by solving the governing equations with parameters and boundary conditions reflecting a real scene. *Fluid control*, on the other hand, is the determination of parameters in a way that the resulting fluid motion follows a prescribed behavior.

Let us consider a fluid element of unit mass. Its motion is described by Newton's second law, stating that the acceleration  $\mathbf{a}$ , which is the derivation of velocity  $\mathbf{u}$  is proportional to the total force:

$$\mathbf{a} = \frac{d\mathbf{u}}{dt} = \mathbf{F}_{\text{int}} + \mathbf{F}_{\text{ext}}$$

where the force is expressed as the sum of internal forces  $\mathbf{F}_{\text{int}}$  and external forces  $\mathbf{F}_{\text{ext}}$ . Taking into account the specific phenomena of fluids, this equation becomes the first Navies-Stokes equation:

$$\frac{\partial \mathbf{u}}{\partial t} + \underbrace{(\mathbf{u} \cdot \nabla) \mathbf{u}}_{\text{advection}} = - \underbrace{\frac{1}{\rho} \nabla p}_{\text{pressure}} + \underbrace{\nu \nabla^2 \mathbf{u}}_{\text{diffusion}} + \underbrace{\mathbf{F}_{\text{ext}}}_{\text{external forces}}, \quad (1)$$

where  $\mathbf{u}(\mathbf{x}, t)$  is the traditional notation of the velocity field in fluid mechanics,  $\rho$  stands for the density,  $p$  for pressure,  $\nu$  denotes the kinematic viscosity of the fluid,  $\mathbf{F}_{\text{ext}}$  is the representation for the sum of all external forces acting on a unit mass fluid element, and the internal force is decomposed to the force of pressure and the internal friction.

There are different ways to approach the numerical simulation of fluids [SS10]. The Lagrangian method treats the fluid as a collection of particles that move in space and time according to their velocities and positions. The Eulerian approach considers the problem from a different point of view: it defines a stationary grid by its points, and the relevant quantities like pressure, velocity and density are measured in these points. The fluid is expected to flow through and between these grid points. The quality of the simulation depends greatly on the resolution of the grid and since it is impossible to compute the quantities in infinite points of the continuum, an interpolation function is used to obtain information from between the sample points. An intuitive example to help understanding the key difference between the two viewpoints is the following: when collecting data for a weather report, the meteorologist can sit in an air balloon and measure quantities like temperature, wind speed and directions, humidity while floating along the flow of the air. This would be the Lagrangian approach. Someone utilizing the Eulerian viewpoint may simply set up stationary measurement devices in various places throughout the country. Advection term  $(\mathbf{u} \cdot \nabla)\mathbf{u}$  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. The first Navier-Stokes equation expresses the conservation of momentum. In addition to this, the second equation enforces the conservation of mass:

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

For a detailed introduction to fluid dynamics we refer the reader to [CM93] and [Bri08].

Computational fluid dynamics enjoys a wide variety of use in a number of engineering and physics problems: it is possible to visualize heat distribution in a newly designed car engine, to validate airplane design by performing wind tunnel tests with a computer software, simulate gas or water flow in chopper pumps, visualize and plan optimum air circulation conditions in buildings, calculate the drag for accelerating vehicles of various shapes, or to understand the possible outcomes better when a catastrophe, such as a flood, the tsunami, or a volcanic eruption happens. There are also methods to help the medical examination of humans by detecting probable spots in the aorta for aneurysms, little bulges in the wall of blood vessels [RIM\*11], which, under extremal pressure conditions, may explode, causing oftentimes lethal implications.

The more rigorous statement of the fluid control problem is the following: there is a given density field that is to converge to a target density in time, while retaining only natural movement in the fluid domain. To make the fluid obey the control, we apply an external force field, intuitively meaning that we have little spoons of infinitesimal size, which we use to stir the fluid in the chosen

directions, doing it several, or even thousands of places at once. To measure how good our job was done with the stirring, we measure the difference between the obtained and the target density field.

Mathematical algorithms for most problems can be objectively compared as they have their own, well-defined error metric. Comparing them is as simple as choosing the one that offers the best time-quality trade-off for our needs. However, there is no metric that would define how natural a fluid flow is, we are left to judge by what *seems* and what *feels* to be natural for us.

In this paper, we present the following contributions:

- Statement of the fluid control problem and a brief overview of a state-of-the-art solution to it.
- Observations regarding the fluid control problem, where simplifications can be made to the target distribution while retaining its applicability to both industrial and artistic use.
- A novel, highly parallelizable method to simplify the optimization approach that relies only on local data, and which is able to solve the fluid control problem in real time while retaining realistic looking fluid flows.
- An implementation of the algorithm on GPU hardware that is released alongside this paper at website<sup>1</sup>.

## 2 Related work

Solving the Navier-Stokes equations requires the evaluation of three terms and keeping the velocity field divergence-free throughout the process, for instance by using Helmholtz-Hodge decomposition. However, the advection operator consists of a directional derivative  $(\mathbf{u} \cdot \nabla)\mathbf{u}$ , where using the most straightforward forward projection operator will result in an unstable simulation. Jos Stam addressed this problem with a stable advection formula in his paper [Sta99], where instead of checking how the current density field would evolve in a time step, he advised projecting the density field backwards and see where the density came from for every point on the grid. Simulating fluids on a finite resolution grid has a serious drawback, resulting in two symptoms: the first is that the simulation of turbulent flows requires the modeling of high frequency changes in the velocity field. Unfortunately, this is quite costly as increasing the grid resolution raises the computational cost of the simulation by the magnitude of  $\Theta(n^d)$  for a simulation of dimension  $d$ . Fedkiw et al. proposed a way to reinject the lost energy into the system in the form of an external force field [FSJ01]. Literature refers to this technique as Vorticity Confinement ( $F_{vc}$ ). The second drawback of the finite grid approach is that velocity information between the grid point is obtained by bi- or trilinear interpolation. Unfortunately, the Navier-Stokes equations are not linear, but parabolic partial differential equations, therefore higher order advection methods, such as the Back and Forth Error Compensation and Correction Advection [DL03] and MacCormack Advection [SFK\*08] yield better

---

<sup>1</sup> <http://cg.iit.bme.hu/~zsolnai>

quality solutions. It is also possible to obtain additional realism with generating flow data with frequency higher than the Shannon-Nyquist limit for the given grid resolution by using the Wavelet Transform [KTJG08].

Several solutions exist to solve the fluid control problem, such as Jos Stam's adjoint method [MTPS04], or the algorithm of Shi and Yu that adds a long-range force field to even out the distribution of the fluid on a macro level, and a short-range field to carve out the fine details of the target distribution [SY04]. Our approach based on this, so we provide the details of the calculation of the long-range and short-range forces.

## 2.1 Long-range force fields

If some part of the fluid domain has excess density, meaning that  $\rho_j$ , the density at point  $j$  is higher than target density  $\rho_j^t$ , the region will transport density by exerting force towards the direction of its neighborhood for those who have lower density than the target. As in physics, the exerted force weakens as the distance between the two points increases, therefore to maintain energy conservation, some kind of falloff characteristic has to be introduced. This will be the definition of the long-range force field in [SY04] that is responsible for arranging the density field so it can start converging to the target on a macro level:

$$\mathbf{F}_i^L = \sum_j \frac{[\rho_j - \rho_j^t]^+}{|\mathbf{r}_{ij}|^\alpha} \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|}, \quad (3)$$

where  $\mathbf{r}_{ij}$  is a vector that points from grid point  $j$  to  $i$ , and superscript  $+$  stands for an operation that replaced negative values by zero. Most physical phenomena such as light propagation, or forces like gravity and electric force have a squared falloff with increasing distance, therefore it is not surprising that the choice of  $\alpha = 2$  gives visually pleasing results.

## 2.2 Short-range force fields

Now a short-range force field is to be constructed to carve out the fine details of the target distribution locally. In [SY04], Shi and Yu have advised that short-range force field  $\mathbf{F}^S$  is obtained as the solution of the following global optimization problem:

$$\begin{aligned} \min_{\mathbf{F}^S} c_1 \sum_{\mathbf{x}} (\Delta\rho(\mathbf{x}) - \lambda\Delta\rho^a(\mathbf{x}))^2 + c_2 \sum_{\mathbf{x}} (DIV(\mathbf{x}))^2 - \\ c_3 \sum_{\mathbf{x}} \left( \frac{\mathbf{F}(\mathbf{x})}{|\mathbf{F}(\mathbf{x})|} \cdot \frac{\nabla\rho(\mathbf{x})}{|\nabla\rho(\mathbf{x})|} \right)^2 - \\ c_4 \sum_{\mathbf{x}} \sum_{\mathbf{y}} \cos(\theta(\mathbf{y}) - \theta(\mathbf{x})) + \\ c_5 \sum_{\mathbf{x}} \sum_{\mathbf{y}} (|\mathbf{F}^S(\mathbf{y})| - |\mathbf{F}^S(\mathbf{x})|)^2. \end{aligned} \quad (4)$$

where every function depends on position  $\mathbf{x}$ . The  $c_1, \dots, c_5$  coefficients are used to assign different weight values to the terms. We attempt to give an intuitive interpretation of the formula: the *first term* is responsible to ensure that the difference between the current and the target density is minimal by maximizing the amount of density change in excess density areas, and minimizing it at areas that match the target density well. The velocity field has to be divergence-free throughout the simulation and control process, therefore in the *second term*, substituting  $\nabla \cdot \mathbf{u}(\mathbf{x})$  into  $DIV(\mathbf{x})$  makes a plausible choice. However, measurements show that we obtain better results by using the long-range force field, evaluate the projection step afterwards to keep the divergence-free property, then using  $\nabla \cdot \mathbf{F}^S(\mathbf{x})$  as a divergence term. The *third term* is the dot product between the normalized direction vectors of the force field and the gradient of the density field. Minimizing this is absolutely important to achieve natural looking flows: the applied forces will be similar to the natural flows in the fluid domain for any given time. As  $\theta(\mathbf{x})$  represents the orientation of the short-range force field in point  $\mathbf{x}$ , the *fourth and fifth terms* are to minimize the amount of directional variance and the magnitude of the applied short-range force field.

In [SY04], the idea of a new diffusion scheme has also been raised, where diffusion, instead of its original role of smoothing the discrepancies of the density field, would rather help the convergence of the image through acting as a stronger distributive force in regions of poor convergence. Our experience shows that using this method excels at controlling the density towards the feature points of the target, therefore it is a valuable tool to obtain convergence for complex setups, but at the cost of losing some realism on the fluid flow as the diffusion process will be conspicuously asymmetric. If target density distribution is of high variance, using biased diffusion helps convergence substantially. If maximizing the realism of the fluid flow is a more important requirement, then omitting this technique is the right decision, alongside with using low-variance target distributions.

### 3 The proposed method

The most important attribute of the new approach would be not to use short-range forces due to its computational costs, therefore omit carving out some of the fine details, but design a different, cheaper force field that is able to mobilize large amounts of density towards the target density while still retaining a realistic looking flow.

Long-range force computation given by equation 3 is also modified. Constructing this force field requires interactions whose number is a quadratic function of the number of grid points, which is undoubtedly too demanding. Luckily, it isn't necessary, since as we increase the distance, every sensible choice of  $\alpha$  will make the force decay in an at least quadratic manner, we can safely define a maximum distance that is to be computed from every point.

In addition, our version we have also relaxed the evaluation of the long-range forces to regions that have nonzero target density,

$$\mathbf{F}_i^L = \sum_j \frac{[\rho_j - \rho_j^t]^+}{|\mathbf{r}_{ij}|^\alpha} \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|}, \quad \forall j : \rho_j^t > 0, \quad (5)$$

to reduce its cost to be proportional to only the volume where the target distribution is nonzero, as opposed to the original method, where it is evaluated in the full simulation domain.

Due to the simplification, it can occur that upon reaching convergence, excess density remains around the outer side of the boundary of the target distribution. To clean up these details around the boundary, an additional force field  $\mathbf{F}^C$  may be applied that sucks the density back into the domain of the target distribution. The use of this cleanup force field is entirely optional, and it would consist of vectors that are oriented from points that contain density but have zero target density and are near the boundary to every other point that has nonzero target density. Formally:

$$\mathbf{F}_i^C = \sum_{j:\rho_j^t>0} \frac{\rho_j}{|\mathbf{r}_{ij}|^\alpha} \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|} \quad \forall i : \rho_i^t = 0. \quad (6)$$

Deciding what should happen after the target density is reached is also a very important task. While it would be straightforward to, for example, set  $\nu$  to a very high value to to “freeze” the fluid in the convergent subdomains, the results will remain correct, though can not be expected to look very lifelike, making it a simple, effective, but generally unconvincing solution.

Here we try to address the shortcomings of this approach by introducing a speedup factor  $\psi$ , which increases the velocity of the fluid at regions where convergence is reached.

It may sound quite counterintuitive: why speed up the fluid at regions where it already looks correct? It would be reasonable to say that it is the exact opposite of what should happen. On a microscopic level, freezing the fluid domain would always give the best choice: if we have only a few points in space, freezing them by assigning a very high kinematic viscosity value upon reaching the target distribution will ensure that they will remain in the correct state all the time. Let’s consider a simple example on a macroscopic level, where we have a fluid domain of significant size where the target distribution can be reached by going through a narrow choke point. At this region, the fluid will start freezing, therefore it will prevent any further fluid movement, making it impossible to get density through. This scenario will not be restricted only to narrow choke points: for almost every practical case on closed shapes, the closer we are to the state of convergence, the higher the probability for this to happen. Intuitively, freezing the fluid can be associated with the “after you’re done, just stop and rest” behavior, as opposed to speeding up the velocity, which would mean more like “after you’re done, start helping others”. This behavior will not only allow the fluid to flow through narrow choke points, but effectively uses them to its

advantage by turning them into local regions that transfer density to neighboring regions of poor convergence. As our results show, this approach will not just preserve fluid movement after the target density is reached, but will help increasing the speed of the convergence as well.

### 3.1 The final form

Putting it all together, the proposed form of the Navier-Stokes equations for controlling the fluid is as follows:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = \psi \left( -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} \right) + \mathbf{F}^L + \mathbf{F}^C + \mathbf{F}_{vc},$$

$$\psi = \begin{cases} 1 + \delta, & \text{where } |\rho^t - \rho| < \epsilon, \\ 1, & \text{elsewhere,} \end{cases} \quad (7)$$

where the forces are the modified long-range force, cleaning-up force and the vorticity confinement force, and  $\delta$  is small value to keep the fluid in motion after convergence when the actual density is equal to the target density within error threshold  $\epsilon$ . The differential equation can be solved with Neumann, e.g.

$$\frac{\partial \mathbf{u}}{\partial n} \Big|_{\partial \Omega} = 0.$$

or Dirichlet type boundary conditions. This technique is capable of guiding the fluid towards the target distribution in real-time.

## 4 Results

The properties of the proposed technique are as follows:

- The fluid simulation and the control algorithm can be run in parallel as they take 18 and 14 milliseconds at most respectively on an  $512^2$  grid with 20 Jacobi iterations, therefore it is a real-time solution,
- It yields fast convergence speeds,
- It is to be used with target distributions of low-variance for a high measure of realism, or,
- It is to be used on more complex, higher-variance distributions with the aid of biased diffusion at the cost of less realism and more computational demands,
- Relying only on local data, it can be extended to simulations of any dimension with favorable amount of computational overhead.

Our test results using Neumann-type boundary conditions without control force fields show that it is unreasonable to expect a desirable degree of convergence

(Figure 1). This method is also unable to control any amount of density which is not already inside the target distribution domain. The results using the new method were rendered in real-time and are shown in Figure 2, which, using roughly the same amount of density, was able to achieve fast convergence.

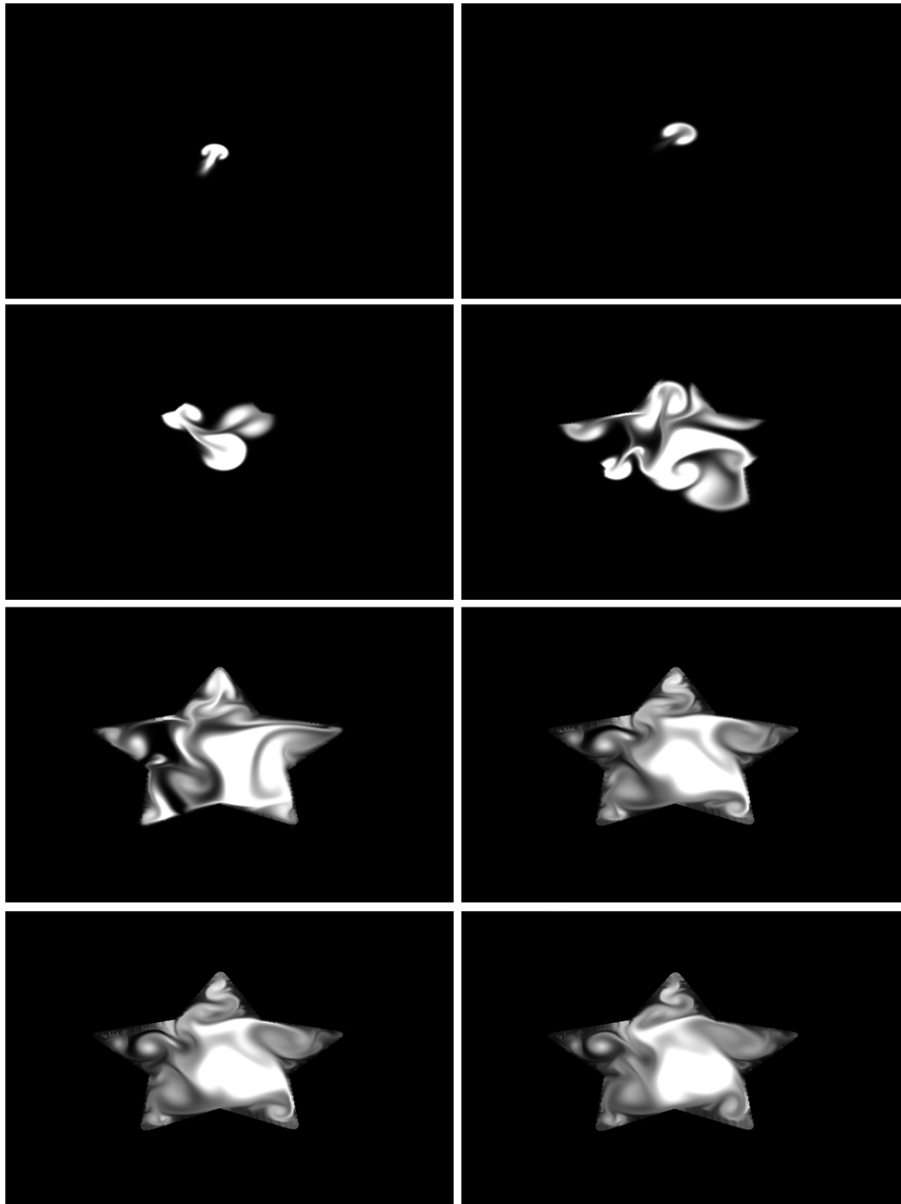
## 5 Conclusions

In this paper, we have shown a novel algorithm to the fluid control problem, where an initial state and a target distribution are given for the fluid, with the intention that it would sooner or later take the form of this distribution. A solution to this problem is an external force field that is changing in time, describing the forces that have to be exerted on the fluid to take a given shape. This force field makes the fluid converge in a short amount of time, with the slightly ambivalent requirement of giving a realistic flow in the meantime, even if it is highly unlikely that a bowl of water would suddenly take the form of natural objects. Unlike state-of-the-art methods giving convincing results at the cost of 5 to 7 minutes of computation time per frame, our method is able to solve the fluid control problem effectively.

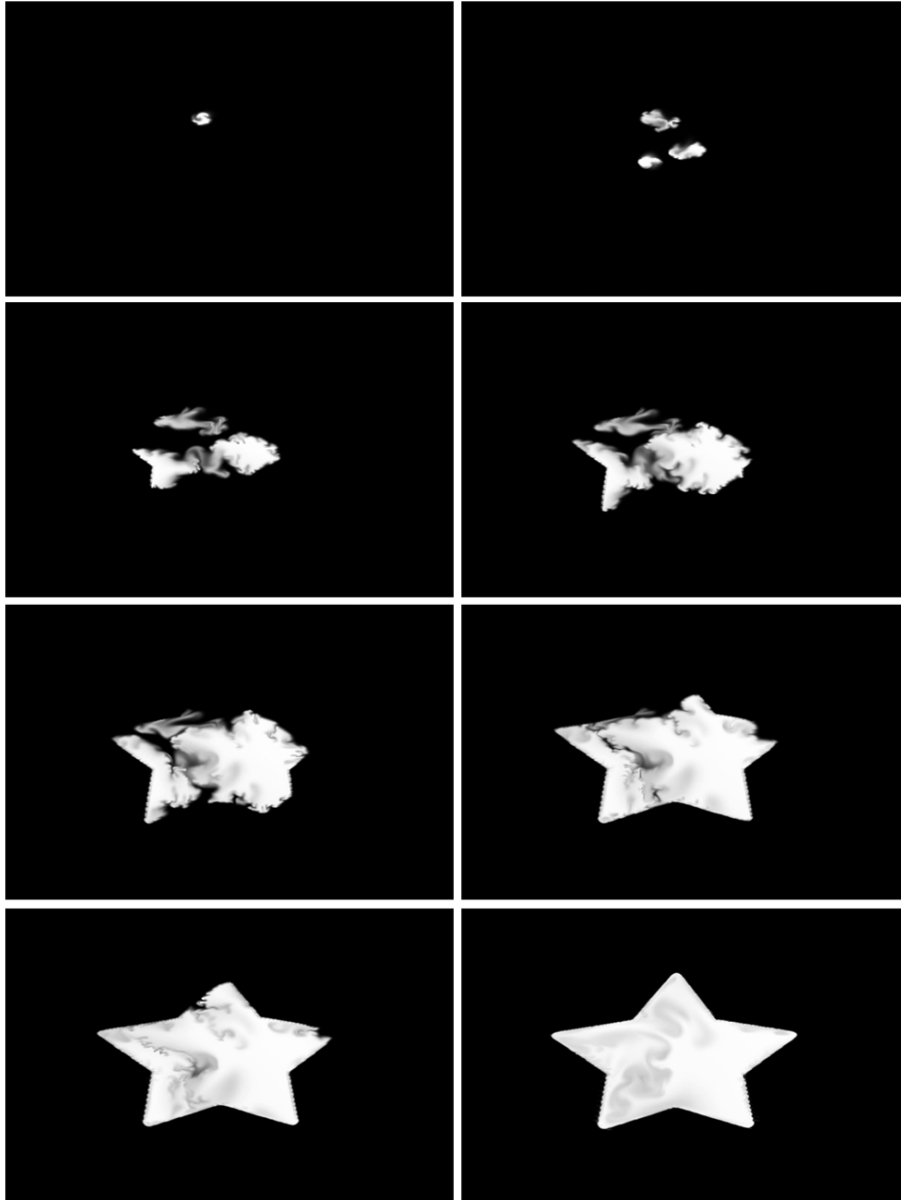
## Acknowledgements

This work is supported by TÁMOP-4.2.2.B-10/1-2010-0009 and OTKA K-104476.





**Fig. 1.** An imitation of the solution to the fluid control problem where no control forces, only boundary conditions are used. 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.



**Fig. 2.** The proposed method runs in real time, provides good coverage of the target density, and is aware of the regions of poor convergence, which are constantly helped out by nearby regions. Roughly the same amount of density is used as in Figure 1.

## References

- [Bri08] BRIDSON R.: *Fluid Simulation*. A. K. Peters, Ltd., Natick, MA, USA, 2008.
- [CM93] CHORIN A., MARSDEN J.: *A Mathematical Introduction to Fluid Mechanics*. Texts in Applied Mathematics. Springer-Verlag, 1993.
- [DL03] DUPONT T. F., LIU Y.: Back and forth error compensation and correction methods for removing errors induced by uneven gradients of the level set function. *J. Comput. Phys.* 190, 1 (Sept. 2003), 311–324.
- [FSJ01] FEDKIW R., STAM J., JENSEN H. W.: Visual simulation of smoke. In *ACM SIGGRAPH 2001* (2001), pp. 15–22.
- [KTJG08] KIM T., THÜREY N., JAMES D., GROSS M.: Wavelet turbulence for fluid simulation. In *ACM SIGGRAPH 2008 papers* (New York, NY, USA, 2008), SIGGRAPH '08, ACM, pp. 50:1–50:6.
- [MTPS04] MCNAMARA A., TREUILLE A., POPOVI Z., STAM J.: Fluid control using the adjoint method. *ACM TRANS. GRAPH. (SIGGRAPH PROC 23)* (2004), 449–456.
- [RIM\*11] RALOVICH K., IONASEC R. I., MIHALEF V., SHARMA P., GEORGESCU B., EVERETT A., NAVAB N., COMANICIU D.: Computational fluid dynamics framework for large-scale simulation in pediatric cardiology. *Computational Biomechanics for Medicine VI (CBM6) MICCAI Workshop* (2011).
- [SFK\*08] SELLE A., FEDKIW R., KIM B., LIU Y., ROSSIGNAC J.: An unconditionally stable maccormack method. *J. Sci. Comput.* 35, 2-3 (2008), 350–371.
- [SS10] SZIRMAY-KALOS L., SZÉCSI L.: General Purpose Computing on Graphics Processing Units. In *Algorithms of Informatics*, MondArt Kiadó (2010).
- [Sta99] STAM J.: Stable fluids. In *Proceedings of SIGGRAPH 99* (1999), Computer Graphics Proceedings, Annual Conference Series, pp. 121–128.
- [SY04] SHI L., YU Y.: Inviscid and incompressible fluid simulation on triangle meshes. *JOURNAL OF COMPUTER ANIMATION AND VIRTUAL WORLDS* 15 (2004), 3–4.