

# A Stable Fluids Approach To Multiple Types Of Smoke

Janus Dam Nielsen\*

University of Aarhus, Denmark

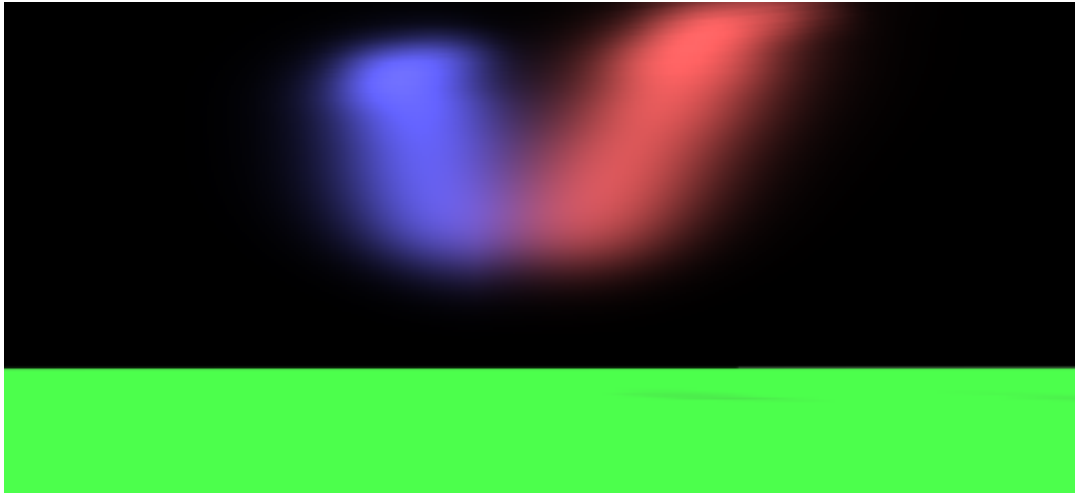


Figure 1: Two types of smoke interacting like the exhaust of aircrafts at aircraft shows.

## Abstract

This paper presents a stable method for simulating different types of smoke in the same scene under the presence of varying temperature and vorticity confinement. Multiple types of smoke are naturally occurring simultaneously in many different situations like smoke from different fires, smoke from different parts of the same fire or in aircraft shows where a number of aircrafts emit different colors of smoke. In these kinds of situations temperature differences and rotational details are often important to get physically realistic and visually appealing smoke. The presented method is based on physical realistic simulation of smoke using semi-Lagrangian advection and the conjugate gradient method to solve the Navier Stokes equations similar to the Stable Fluids method and extends the Stable Fluids method with multiple types of smoke moving in the same velocity field, a temperature field, and vorticity confinement for more realistic simulations of turbulent flows. The method adds three techniques to the Stable Fluids method which gives the possibility of simulating multiple types of smoke realistically. The results produced by combining the additional techniques are visually satisfactory and realistic. Results obtained when comparing only each of the additional techniques are comparable to the results obtained by others.

**CR Categories:** : Three-Dimensional Graphics and Realism—Animation; Curve, Surface, Solid and Object Representations; : Computational Geometry and Object Modeling—Physically Based Modeling; : Types of Simulation—Animation

**Keywords:** Computational fluid dynamics.

## 1 Introduction

Multiple types of smoke are naturally occurring and interacting in many situations like action films where multiple sources of fire often appear in the same scene, it could be a barrel of gasoline at fire emitting a very thick heavy smoke combined with the smoke from burning cars or buildings, or flares often used to mark helicopter landing zones. Different types of smoke may also be emitted by fires covering large areas of heterogeneous material which burns with different temperatures and emits different types of smoke with different color and viscosity. Different types of smoke are also popular at aircraft shows where aircrafts flying in acrobatic formations often emit smoke with different colors which mix in various ways.

Simulating multiple types of smoke occurring in the same scene, in a physically realistic way would allow the simulations of the above cases and so increase the realism and diversity of the scenarios. This paper presents a stable physics-based method specifically designed to realistically simulate multiple types smoke.

Physics-based simulation of smoke has been widely studied both in computer graphics and in engineering. Fluid mechanics is often used as the mathematical basis for these simulations and especially the Navier Stokes equations are used as a model for fluids. In computer graphics some lack of physical accuracy is often accepted for visually appealing effects, but the effects should in general behave physically correct. The Stable Fluids method by Stam [Stam 1999] is a method for solving the Navier Stokes equations in a stable way, where the “blow up” experienced in other methods is avoided and results in a physically correct simulation to the extended needed. The stability combined with physical correctness makes it a good foundation for realistic simulation of multiple types of smoke and is used as a basis for the method presented in this paper.

In the Stable Fluids method simulation takes place within a confined discrete simulation domain and the single type of smoke supported by the method is described by a discrete scalar field of smoke

---

\*e-mail: jdn@brics.dk, Aarskort-nummer: 20000931

density which is evolved according to the Navier Stokes equations. In the model presented in this paper different types of smoke are represented in similar way as a discrete density field which are evolved according to the Navier Stokes equation.

The realism of the simulation can be increased by taking temperature into account. In action films and aircraft shows the heat of the smoke affects the flow of the smoke in important ways e.g. in aircraft shows the smoke is emitted at the back of the aircraft often close to the exhaust of the jet engine which is many orders of magnitude more warm than the surrounding air and so makes the colored smoke rise rather quickly. The method presented in this paper simulates the effects of different temperatures in the simulation domain in a similar way as done by Fedkiw et al [Fedkiw et al. 2001].

The physical correctness of simulations based on the Stable Fluids method can be increased by applying vorticity confinement which reintroduces the small scale rotational details lost due to numerical dissipation. Vorticity confinement has been implemented based on the work by Fedkiw et al [Fedkiw et al. 2001].

The presented method has mainly been developed with the aim of simulating different colors of smoke from multiple aircrafts and has mainly been evaluated on this scenario all though the method should be applicable to many other situations.

## 2 Contributions and Outline of Paper

This paper has three distinct contributions compared to previous work, Stam [Stam 1999] and Fedkiw et al [Fedkiw et al. 2001]. The presented method adds:

- Support for multiple types of non-affecting smoke where the different types of smoke are simulated together in the same velocity field, but do not affect each other.
- Support for simulation of temperature effects using a temperature buoyancy force based on a temperature field similar to the work by Fedkiw et al. [Fedkiw et al. 2001].
- Support for vorticity confinement where small scale details lost due to numerical dissipation are reintroduced using an additional force term, the vorticity force, based on how much the velocity field rotates at each point of the domain similar to the work by Fedkiw et al. [Fedkiw et al. 2001].

The contributions add both to the domain of what is possible to simulate using the Stable Fluids method, multiple types of smoke and temperature effects, and to the realism of the simulations, vorticity confinement. In addition to the contributions the presented method also include diffusion and dissipation of smoke similar to [Stam 1999].

The rest of the paper is organized as follows. In Section 3 a short description of related work is given. Section 4 gives the necessary physical background for the method presented in Section 5 where a thorough presentation of the method is given. In Section 6 the results obtained using the presented method are discussed and

compared to previous work. The paper conclude in Section 7 with future work and conclusion.

## 3 Previous Work

The Stable Fluids method is introduced by Jos Stam in the papers [Stam 1999; Stam 2003] and is the foundation on which the method presented in this paper is build. The Stable Fluids method is based on the key observation that the equations describing the physical laws of fluid dynamics, the Navier Stokes equations, can be solved by tracing the various quantities backwards in time with proper boundary conditions and hence gives unconditional stability since the maximum value of each field to a time,  $t + \Delta t$ , can not be greater than the maximum value of the field at the earlier time  $t$ .

Vorticity confinement adds back to the smoke the small scale rotational effects lost due to numerical dissipation, using a force term proportional to the rotation of the velocity field. In this paper vorticity confinement is presented and used to add realism to animations of multiple types of smoke. The work is based on the work by Fedkiw et al [Fedkiw et al. 2001] but Fedkiw et al stores velocities at voxel faces whereas the method presented below stores velocities at voxel centers.

How to animate smoke under influence of evolving temperatures throughout the simulation domain has been studied by Fedkiw et al [Fedkiw et al. 2001] and Foster and Metaxas [Foster and Metaxas 1997] where the temperature in the simulation domain is represented as an additional scalar field which is evolved in the same way as the smoke density field. The effect of the temperature on the smoke is modeled using a buoyancy force which is proportional to the difference between the temperature at each point and the ambient temperature of the domain or obstacle.

## 4 Equations of Fluid Flow

This section introduces the necessary physical foundation and understanding to fully appreciate the presented method. This section relies on the reader having some mathematical knowledge and physical intuition.

The simulation takes place within a confined volume of space, the simulation domain. The simulation domain is discretized into a number of voxels each separated by the same distance  $\Delta x$  in all three dimensions. The smoke is, in terms of physics, a fluid and evolves according to the laws of physics inside the simulation domain. The smoke is assumed to have a near constant density,  $\rho$  - mass per volume, over the entire domain. The evolution of the smoke can be described using a velocity field,  $\mathbf{u}$ , and a number of scalar quantity fields,  $q$ . A scalar quantity field can be used to describe various physical quantities like smoke density and temperature. The same physical laws describe the evolution of these quantities and so they are treated together. Each field describes the corresponding quantity at the center of each voxel of the domain. The values of a field generally varies in both time and space and depends on the boundaries of the simulation domain.

The evolution of the quantity fields depends on the velocity field along with diffusion and dissipation. Diffusion happens when quantities moves from areas of high concentration to areas of low concentration. Dissipation is due to loss of energy. The evolution of the velocity field depends on itself and the forces applied to it like buoyancy (from the temperature) and gravity. These complex relationships are captured in the equation for evolution of a quantity

like smoke densities 1 and the *Navier Stokes* equations 2 and 3. The smoke is assumed to preserve momentum, 2 and mass 3.

$$\frac{\partial q}{\partial t} = -\mathbf{u} \cdot \nabla q + \kappa \nabla^2 q + S_q + \alpha q \quad (1)$$

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

$$\nabla \cdot \mathbf{u} = 0 \quad (3)$$

Given an initial quantity to the initial time  $t = 0$  and the velocity at each time  $t$ , the *quantity* equation 1 describes how the quantity, e.g. the smoke density, evolves over time. Each term on the right-hand side describes how the evolution depends on various parameters. The *advection* term,  $-\mathbf{u} \cdot \nabla q$ , describes how the density evolves as a function of the velocity. The *diffusion* term,  $\kappa \nabla^2 q$ , where  $\kappa \geq 0$  is the diffusion rate, describes the net change in density due to diffusion. The *source* term,  $S_q$ , tells how much new smoke should be added e.g. in the case of an aircraft the source would be the back of the aircraft. The last term is the dissipation term  $\alpha q$ , where  $\alpha \geq 0$  is the dissipation rate, which describes how much smoke should vanish due to dissipation.

Given some initial velocity and pressure at time  $t = 0$ , the Navier Stokes equations describe how the velocity evolves over time. The similarity of equation 1 and 2 suggest that equation 2 can be broken down into similar terms as equation 1. The term  $-(\mathbf{u} \cdot \nabla) \mathbf{u}$  is called the *advection* term and describes how the velocity field evolves as a function of itself - the velocity is moved by itself. The term  $\nu \nabla^2 \mathbf{u}$ , is the *diffusion* term, where  $\nu$  is the diffusion or viscosity coefficient, and describes how the fluid should deform under shear stress. The term  $\mathbf{f}$  is the *force* term, which is similar to the source term in equation 1 and simply adds force to the velocity. The term  $-\nabla p$  is the *pressure* term and describes the pressure of the fluid.

In addition to the equations sufficient boundary conditions are needed. In this paper two different kinds of boundary conditions are used. Von Neumann boundary conditions where the value just outside the boundary is the same as just inside the boundary and Dirichlet boundary conditions where the boundary condition is described by some function, a constant zero function is used. Furthermore the normal of the velocity is set to zero at the boundaries such that no matter escapes the simulation domain.

Three distinct forces influence the evolution of the velocity in the presented method and so a short tour de force is given through the gravity, buoyancy, and vorticity forces. The gravity forces at a given point is directed towards the ground (negative along the y-axis) and is directly proportional to the amount of mass present at the point and so proportional to the total density,  $\rho_{total}$ , of smoke.  $\alpha > 0$  is the gravitational constant.

$$\mathbf{f}_{gravity} = -\alpha \rho_{total} \mathbf{y} \quad (4)$$

The buoyancy force [Fedkiw et al. 2001] is due to differences in temperature in the simulation domain and only affects the velocity along the y-axis. The buoyancy force is directly proportional to the difference between the temperature and the ambient temperature of the air.  $\beta > 0$  is the buoyancy constant.

$$\mathbf{f}_{buoyancy} = \beta (T - T_{amb}) \mathbf{y} \quad (5)$$

Vorticity confinement [Fedkiw et al. 2001] adds back the small scale turbulence lost due to numerical dissipation by applying additional force to the velocity at places where small rotations of the velocity already occurs and so enhance the rotations to their proper

strength. The vorticity,  $\omega$ , at a given point is calculated as the curl of the velocity field at the given point.

$$\omega = \nabla \times \mathbf{u} \quad (6)$$

The additional vorticity force,  $\mathbf{f}_{conf}$ , at a given point is the cross product of the normalized vorticity location vector and the vorticity.

$$\mathbf{N} = \frac{\eta}{|\eta|} (\eta = \nabla |\omega|) \quad (7)$$

$$\mathbf{f}_{conf} = \varepsilon \Delta x (\mathbf{N} \times \omega) \quad (8)$$

Where  $\varepsilon > 0$  is the vorticity coefficient describing the amount of small scale rotation which should be added back into the scene and  $\Delta x$  ensures the correct proportionality of the force is conserved as the size of the grid changes.

All together the above forces add up to the following total force which must be applied to the velocity at each point:

$$\mathbf{f} = \frac{\partial \mathbf{u}}{\partial t} \quad (9)$$

$$\mathbf{f} = \mathbf{f}_{gravity} + \mathbf{f}_{buoyancy} + \mathbf{f}_{conf} \quad (10)$$

## 5 Multiple Types Of Smoke

This section presents an approach to simulation of multiple types of smoke combined with vorticity confinement under influence of temperature. The method is based on the Stable Fluids method by Stam [Stam 1999]. The method is a combination of the four parts: *the basic Stable Fluids*, *multiple types of smoke*, *temperature*, and *vorticity confinement*. Each of the parts will be described in details below and the section ends with a description of how the parts fit together.

### Stable Fluids - The basics

The Stable Fluids method is a method for solving the Navier Stokes, equations 2 and 3, together with equation 1 and forms the basis for the presented method. At the core of the Stable Fluids method is the crucial observation which makes the method unconditional stable. At each step of the method the value of the velocity field or a quantity field to the time  $t + \Delta t$  is traced backwards in time to its value at time  $t$ .

The method is furthermore based on an important observation known as the Helmholtz-Hodge decomposition theorem which states that any vector field,  $\mathbf{w}$ , can be uniquely decomposed into the sum of a mass conserving field and a gradient field. The Helmholtz-Hodge decomposition theorem allows us to rewrite the Navier Stokes equations into a slightly more tractable form. The Navier Stokes equations describes a mass conserving,  $\nabla \cdot \mathbf{u} = 0$ , vector field  $\mathbf{u}$ , which involves a pressure term and so would require us to represent the pressure explicitly during simulation. A vector field,  $\mathbf{w}$ , can be constructed as the sum of the mass conserving vector field,  $\mathbf{u}$ , and the gradient of the pressure,  $p$ , using the Helmholtz-Hodge decomposition theorem.

$$\mathbf{w} = \mathbf{u} + \nabla p \quad (11)$$

$$\frac{\partial \mathbf{w}}{\partial t} = -(\mathbf{w} \cdot \nabla) \mathbf{w} + \nu \nabla^2 \mathbf{w} + \mathbf{f} \quad (12)$$

The evolution of the velocity field  $\mathbf{w}$  can be solved without taking the pressure into account. The pressure must however be determined from the velocity field,  $\mathbf{w}$ , if the velocity field,  $\mathbf{u}$ , needed in

order to solve to Navier Stokes equations is to be obtained. Fortunately the pressure is straight forwardly found by applying the divergence operator to each side of equation 11 yielding the Poisson equation:

$$\nabla \cdot \mathbf{w} = \nabla \cdot \mathbf{u} + \nabla \cdot \nabla p \quad (13)$$

$$\nabla \cdot \mathbf{w} = \nabla^2 p \quad (14)$$

From which the pressure can be found by expanding and rearranging the terms into a sparse system of symmetric linear equations using finite differencing which should be solved backwards in time to achieve unconditional stability. The most natural linear solver in this case is a conjugate gradient solver [Golub and Loan 1989] with Von Neumann boundary conditions except at a single voxel where a Dirichlet boundary condition of zero is used in order to allow numerical dissipation. The solver is easy to implement and is used with an incomplete Choleski pre-conditioned to improve convergence.

The above rewrite leaves only the task of solving equation 11, which can be done by starting from some initial velocity  $\mathbf{w}_0$  and sequentially resolve each of the terms of the right hand side into an intermediate velocity field  $\mathbf{w}_i$  used as input for solving the next term as in [Stam 1999]. The terms are resolved in the steps shown below: First forces are added, then move the velocity field under influence of itself, advection, and then diffusion of the velocity yielding the final velocity field  $\mathbf{w}_3$ .

$$\mathbf{w}_0 \xrightarrow{\text{add forces}} \mathbf{w}_1 \xrightarrow{\text{diffuse}} \mathbf{w}_2 \xrightarrow{\text{project}} \mathbf{w}_3 \xrightarrow{\text{advect}} \mathbf{w}_4 \xrightarrow{\text{project}} \mathbf{w}_5 \quad (15)$$

The *projection* steps resolve the mass conserving velocity field as described above and is placed before and after the *advection* step because the advection step should behave more accurately when the velocity field is mass conserving according to Stam [Stam 1999].

The addition of forces is easily solved if the forces are assumed not to change considerably during a time-step, which is not unreasonable with the kind and size of forces applied in the presented method:

$$\mathbf{w}_1 = \mathbf{w}_0 + \Delta t \mathbf{f} \quad (16)$$

The advection step calculates the new value of the field as a function of the old value and the velocity. Semi-Lagrangian advection is used to backtrack the velocity at a given time  $t + \Delta t$  to the velocity at time  $t$ , with Dirichlet boundary conditions of zero outside the simulation domain. The backtracking ensures that the maximum value of the field at time  $t + \Delta t$  is not greater than the maximum value of the field at time  $t$ . The backtracking can be done using different approaches combined with trilinear interpolation. The first order accurate backwards Euler integration time-step is simple but

$$q^{t+\Delta t}(\mathbf{p}) = q^t [\mathbf{p} - \mathbf{V}^t(\mathbf{p})\Delta t] \quad (17)$$

not as accurate as the second order accurate backwards Runge Kutta integration time-step which is more complex but more accurate.

$$q^{t+\Delta t}(\mathbf{p}) = q^t \left[ \mathbf{p} - \frac{\Delta t}{2} (\mathbf{V}^t(\mathbf{p}) + \mathbf{V}^t(\mathbf{p} - \mathbf{V}^t(\mathbf{p})\Delta t)) \right] \quad (18)$$

Both approaches are supported by the method and have been implemented but no experiments comparing their quality have been carried out.

The diffusion step calculates how much the velocity at a given point is influenced by the velocity of its neighbour voxels such that voxels with high velocity tends to increase the velocity of low velocity

neighbour voxels. The velocity diffusion is modeled by the heat equation,  $\nu \nabla^2 \mathbf{u}$ , where  $\nu$  is the viscosity and  $\nabla^2 \mathbf{u}$  is the laplacian of the velocity. The heat equation can be rearranged into a system of sparse linear equations which are solved similarly to the pressure using an incomplete pre-conditioned conjugate gradient method.

Equation 1 is solved in similar steps, advection, diffusion and dissipation. Dissipation is just a question of reducing the amount of the quantity,  $q$ , at each point of the field by the dissipation rate, but to ensure stability it must be solved backwards in time:  $q^{t+\Delta t} = \frac{q^t}{(1+\Delta t\alpha)}$ .

## Multiple types of smoke

The presented method extends the Stable Fluids method by allowing any number of non-interacting types of smoke. Each type of smoke has its own viscosity, dissipation rate, and color. The Stable Fluids method handles only one type of smoke which is represented using a density field. The Stable Fluids method generalizes straight forwardly to any number of types of smoke. In the presented model each type of smoke is represented using a separate density field similar to the Stable Fluids method, but the density fields evolves in the same simulation domain under the influence of the same velocity field. Each of the density fields evolves in the same way as described above for the Stable Fluids method and they are handled sequentially in arbitrary order. The velocity field does not evolve during the evolution of the quantity fields. The density fields do not interact directly, however the density fields have indirect influence on each other due to the changes in the velocity field created by the gravity force.

Although the method supports any number of different types of smoke the prototype implementation only support two different types of smoke, this limitation can easily be lifted.

A basic raytracer extended with a ray marching algorithm [Dutre et al. 2002] is used for rendering the simulations produced by the method. The basic raytracer is designed to load a single density field along with a base color for the density field and render the density field similar to [Dutre et al. 2002]. The ray marching algorithm used must however be extended in order to display multiple types of smoke correctly with individual colors. The basic raytracer in the prototype implementation is extended with infrastructure to load two density fields with individual colors. The presented method works for any number of smoke types, as mentioned above, but for simplicity only two types of smoke are supported in the current implementation. Large part of the raytracer infrastructure can be reused since the density fields are assumed to be of the same size and so only the ray marching algorithm needs to be modified. The ray marching algorithm is the same as in [Dutre et al. 2002] except that the density used to calculate the extinction coefficient at a given point of the simulation domain is not only determined by a single density field, but is the sum of the densities in each field at that point. The radiance at each step of the marching algorithm must be influenced by the color of each type of smoke proportional to the smoke density. In the presented method each of the three color components, red, green, and blue has been calculated according to the following equation and added to resulting radiance.

$$color = \sum_{i=1}^N \frac{c_i \rho_i}{N} \quad (19)$$

This approach has been chosen because it gives visually realistic and acceptable results.

## Temperature

Temperature adds a buoyancy effect to the smoke where hot smoke rises upwards and cold smoke falls towards the ground. The temperature in the simulation domain is represented using a discrete scalar field similar to the smoke density fields with the same size and grid spacing. Heat behaves similar to smoke densities in many respects: it tends to diffuse from areas of high temperature to areas of low temperature, it is moved by the velocities in the domain, and it dissipates to some extent. These similarities makes it straight forward to use the same method for simulating the evolution of the temperature as for the evolution of smoke densities. The only difference is the definition of new temperature diffusion and dissipation rates and that the evolution of the temperature is done before the evolution of smoke densities.

During the *add force* step the buoyancy force,  $\mathbf{f}_{buoyancy}$  must be calculated at each point,  $\mathbf{p}$ . Equation 5 describes how the buoyancy force depends on the temperature difference between the temperature at the given point and the ambient temperature. Calibration of the upwards buoyancy against the downward gravity can be a complicated task with great effect for realism of the simulation. Calibration becomes complicated if the temperature is stored directly in the temperature field because the temperatures must be updated consistently to maintain their relative values. Such updates can be a frequent source of error, therefor are the actual values stored in the temperature field between zero and one and then multiplied by a temperature factor  $\gamma$  when calculating the buoyancy force.

$$\mathbf{f}_{buoyancy}(\mathbf{p}) = \beta(\gamma T_{\mathbf{p}} - T_{amb})\mathbf{y} \quad (20)$$

In this way the temperature factor can be adjusted to find the proper balance with the gravity without changing the values of the field and hence eliminates a potential source of error.

The buoyancy force only acts in the upwards direction and is straight forwardly added to the other forces present in the simulation.

## Vorticity confinement

Vorticity confinement adds small scale rotational perturbations to the velocity by introducing the vorticity force. The vorticity force at each point of the simulation domain is calculated as part of the *add force* step. The vorticity force is calculated as described in Section 4. First, the vorticity at each point is calculated as the length of the curl and stored in an auxiliary scalar field. Second, the gradient of the vorticity is calculated using Von Neumann boundary conditions which is used to calculate the normal,  $\mathbf{N}$ . Third, the vorticity force is calculated as the product of the vorticity factor, voxel spacing, and the cross product of the normal and the vorticity. The resulting vorticity force is simply added to total force as show in equation 10.

## The combined method

The presented method is a combination of the above four overlapping parts. The method extends the Stable Fluids method by handling temperature, multiple types of smoke and vorticity confinement. The temperature and multiple types of smoke are treated in the same way as the smoke density field in the Stable Fluids method as shown in Figure 2, which shows the individual steps of the algorithm executed at each time-step of the simulation. The velocity field is evolved as in the Stable Fluids method followed by the evolution of the temperature field, which is followed by evolution of the different types of smoke, evolved in arbitrary order. Each field in

```
// First handle velocity
add forces
swap fields
diffuse velocity
swap fields
project velocity
swap fields
advect velocity
swap fields
project velocity
swap fields

// Handle temperature
add sources
swap fields
diffuse temperature
swap fields
advect temperature
swap fields
dissipate temperature
swap fields

// Handle smoke
for each type of smoke do {
  add sources
  swap fields
  diffuse smoke density
  swap fields
  advect smoke density
  swap fields
  dissipate smoke density
  swap fields
}
```

Figure 2: An algorithm for the presented method executed at each time-step.

the simulation is represented by two voxel grids which are swapped after each step of the method. The smoke density fields resulting from each iteration of the algorithm in Figure 2 are saved to disk files and combined during ray tracing to one single frame.

Although dissipation is implemented it is not used in the prototype and is only included in the overview for completeness.

## 6 Results and Discussion

This section contains several examples of smoke simulations. The simulations and the rendering have been run on a dual Intel Xeon 3.06 Ghz machine with 2Gb of ram. The size of the simulation domain varies among the simulations, Figures 3, 4, 5, 6, and 7 have a size of (x 50, y 20, z 35) whereas the Figures 9 and 8 have a size of (x 64, y 63, z 49). All the simulations have been rendered using a ray tracer resulting in 500 by 500 pixel frames, some of the figures have been resized to fit into this paper nicely. All results are obtained using a first order Euler integration time-step during advection of velocity and scalar fields and no dissipation is applied.

Figure 8 is the initial image of a simulation using only the Stable Fluids method and is included to shown that the used raytracer produce correct renderings of acceptable quality. Figure 9 show the same bunny after approx. 40 iteration in a hot temperature field of 53 degrees initially occupying the floor of the simulation domain and shows that the temperature does indeed induce a buoyancy ef-

fect on the smoke, no gravity is present.



Figure 3: Smoke walls after 241 frames.

Figure 6 and 3 are from the smoke walls demo and shows how two types of smoke evolves and interact during a simulation. The simulation is based on two rows of smoke emitters which produce an constant amount of smoke for each frame. The smoke drops towards the ground due to the gravity. The emitters stops after 120 frames and the smoke is left to evolve as described above. No temperature was applied in the simulation. Figure 6 shows the simulation after 20 frames and Figure 3 shows the simulation after 241 frames where it is clear that the two types of smoke have mixed to some degree, e.g. the blue smoke is creeping along the ground and is being mixed with the orange smoke. Figure 7 is another example of two walls of smoke and it is clear that the two types of smoke has mixed in between the two walls.

Figure 4 and 5 shows how smoke from aircrafts at air shows may be simulated. The simulation consists of two parallel smoke sources moving along a straight line as a function of time emitting a constant amount of smoke. Figure 4 shows the simulation after 30 frames, where the two types of smoke have just begun to mix at the back of the smoke lines due to diffusion. Figure 5 shows the simulation after another 10 frames where the mixing of smoke is more clear.

## 7 Conclusion and Future Work

The objective of this paper was to create a method for simulating multiple types of smoke as they often appear in aircraft shows.

The objective has been achieved by extending the Stable Fluids method to handle multiple types of smoke and a temperature field. The method has furthermore been improved by reintroducing the small scale rotational details lost due to numerical dissipation in the Stable Fluids method using vorticity confinement. The resulting method is stable and physically realistic simulations of multiple types of smoke in the same scene.

The presented method can be used as basis for further experimentations with cool scenarios involving multiple types of smoke with different viscosity and dissipation rates to explore the full potential of the method. An interesting future subject also involves different ways to conveniently specify the motion of smoke using interactive tools e.g. describe the path taken by the various aircrafts in an aircraft show in order to easily describe complex flight paths like loops.

A not so related area of future work might be simulation and rendering of mist due to heat over e.g. a radiator or as seen near the horizon on a hot summer day. The mist causes the light to refract according to the refraction index of the hot air. The refraction of light posses a challenge if rendering is done from back to front like in the ray marcher especially when marching towards a light source because the light might in general have arrived along some very distorted path depending on the temperature field.

To conclude, the presented method makes it possible to make physically realistic simulations of multiple types of smoke with vorticity confinement under the influence of temperature.

## 8 Acknowledgments

Thanks to Michael Bang for answering an abundant number of questions even at odd hours. Thanks to Allan Rasmusson for inspiring discussions. Thanks to Aske Simon Christensen for last minute inspiring comments for the animation.

## References

- DUTRE, P., BALA, K., AND BEKAERT, P. 2002. *Advanced Global Illumination*. A. K. Peters, Ltd., Natick, MA, USA.
- FEDKIW, R., STAM, J., AND JENSEN, H. W. 2001. Visual simulation of smoke. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, 15–22.
- FOSTER, N., AND METAXAS, D. 1997. Modeling the motion of a hot, turbulent gas. *Computer Graphics 31*, Annual Conference Series, 181–188.
- GOLUB, G., AND LOAN, C. V. 1989. *Matrix Computations*. The John Hopkins University Press, Baltimore.
- STAM, J. 1999. Stable fluids. In *Proceedings of SIGGRAPH 99*, Computer Graphics Proceedings, Annual Conference Series, 121–128.
- STAM, J., 2003. Real-time fluid dynamics for games.

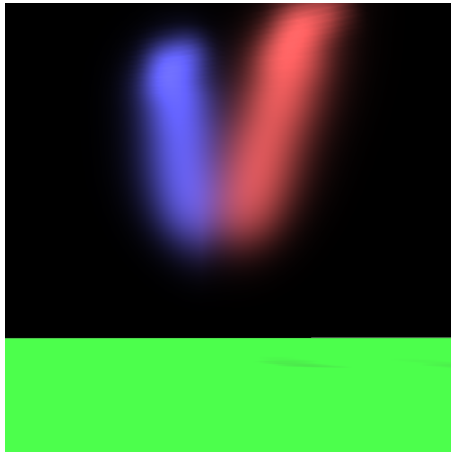


Figure 4: Two types of smoke interacting after 30 frames.

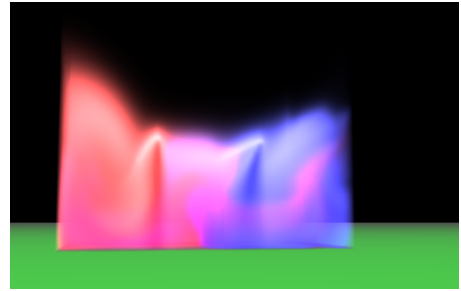


Figure 7: Example of multiple types of smoke interacting.

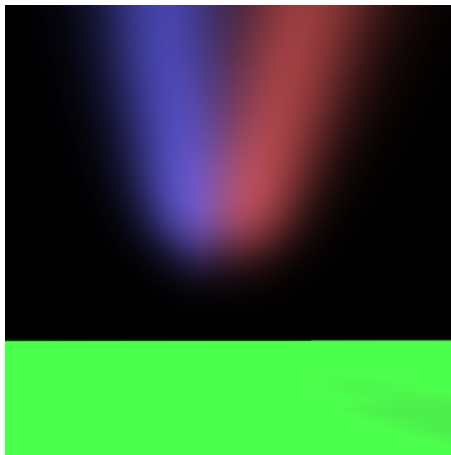


Figure 5: Two types of smoke interacting after 40 frames.

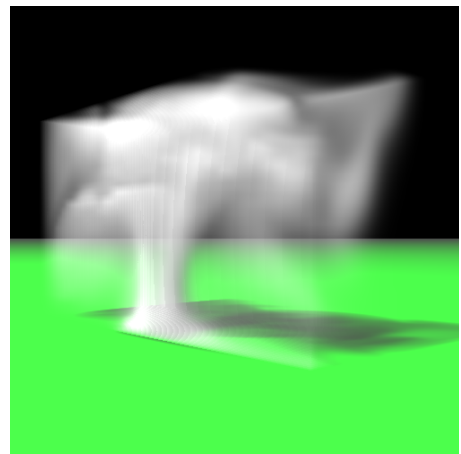


Figure 8: A cloud of grey smoke.

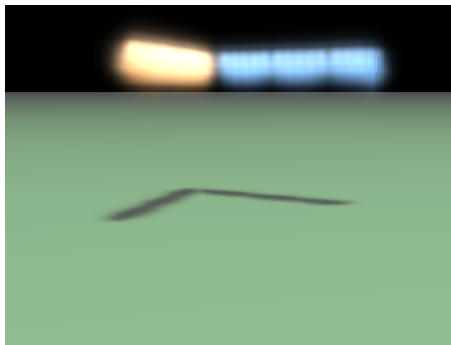


Figure 6: Smoke walls after 20 frames.

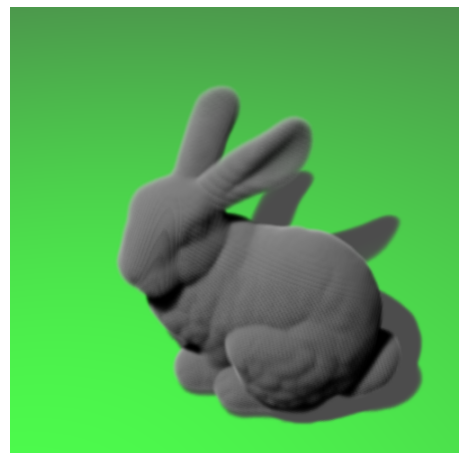


Figure 9: Image of smoke bunny showing that the raytracer works as intended.