Improved Variational Guiding of Smoke Animations

Michael B. Nielsen† and Brian B. Christensen‡

Aarhus University, Denmark

Abstract
Smoke animations are hard to art-direct because simple changes in parameters such as simulation resolution often lead to unpredictable changes in the final result. Previous work has addressed this problem with a guiding approach which couples low-resolution simulations – that exhibit the desired flow and behaviour – to the final, high-resolution simulation. This is done in such a way that the desired low frequency features are to some extent preserved in the high-resolution simulation. However, the steady (i.e. constant) guiding used often leads to a lack of sufficiently high detail, and employing time-dependent guiding is expensive because the matrix of the resulting set of equations needs to be recomputed at every iteration. We propose an improved mathematical model for Eulerian-based simulations which is better suited for dynamic, time-dependent guiding of smoke animations through a novel variational coupling of the low- and high-resolution simulations. Our model results in a matrix that does not require re-computation when the guiding changes over time, and hence we can employ time-dependent guiding more efficiently both in terms of storage and computational requirements. We demonstrate that time-dependent guiding allows for more high frequency detail to develop without losing correspondence to the low resolution simulation. Furthermore, we explore various artistic effects made possible by time-dependent guiding.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.7]: Animation—

1. Introduction
Creating believable fluid effects for computer graphics typically requires the use of advanced and time-consuming computational techniques. For many years the exploration and development of such techniques have been the subject of the computational fluid dynamics (CFD) community. However, whereas numerical and physical accuracy are paramount in CFD, visual plausibility [FSJ01], low simulation cost [Sta99] and artistic control [FM97] are important for computer graphics.

Today, high-resolution fluid simulations that contain sufficient detail for visual effects have turnaround times of several hours or even days. Such simulation times are infeasible for effects-artists during the design stage where near-interactive feedback times are required. For this reason artists typically prototype simulations at low resolution. Once a certain design has been settled for, they increase the resolution to obtain the level of detail required for the shot. However, this approach is not always successful as a change in resolution may not only incur an addition of high frequency features but may completely change the low frequency properties of the flow that the artist settled for in low resolution. Numerical viscosity originating from the discretization is one of the primary reasons for this. This has led several authors to propose algorithms for obtaining better correspondence between low- and high-resolution simulations. This has been done by allowing low-resolution simulations to guide higher resolution simulations [BMWG07, TKPR06, NCZ∗09] and by ensuring that properties such as kinetic energy are preserved in the discretization [MCP∗09].

In this paper we explore time-dependent – as opposed to steady – guiding of smoke animations. Specifically we show that time-dependent guiding can be used for creating dynamic artistic effects and allows for more high frequency detail to develop in high resolution compared to previous work [NCZ∗09]. We take an approach similar to that of Nielsen et al. [NCZ∗09], where guiding is formulated as a
constrained minimization problem. However, Nielsen et al. studied only steady guiding effects and it turns out that their mathematical formulation requires the matrix resulting from their Euler-Lagrange equations to be re-computed, whenever the guiding strength varies in time. In particular, the spatially varying scalar field used to control the guiding strength is by construction included in their matrix. Since this scalar field may change in large fractions of the domain in the case of time-dependent guiding, their matrix re-computation becomes costly when combined with the large stencils required by guiding. Typically their matrix multiplication in the presence of spatially varying guiding weights is about an order of magnitude longer than the simulation time per frame. In addition their matrix storage requirements depend on the variation in the scalar field of guiding weights.

We propose a new mathematical model for guiding smoke animations at high resolution by means of animations at lower resolution. Our mathematical model separates low frequencies from high frequencies in the flow which leads to a set of Euler-Lagrange equations in which the resulting matrix is independent of the scalar field used to control the guiding strength. This means that a costly matrix re-computation does not have to be performed whenever the guiding changes in time. In addition our model often takes about an order of magnitude less time to compute and requires an order of magnitude less storage than [NCZ’09].

Our framework is based on the inviscid Euler equations \( \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} = -\nabla p + \mathbf{f} \) and \( \nabla \cdot \mathbf{v} = 0 \), where \( \mathbf{v} \) is the velocity of the fluid, \( p \) denotes pressure, and \( \mathbf{f} \) represents external forces. We solve these equations using the operator splitting approximation first introduced to computer graphics by Stam [Sta99]. This method solves for self-advection, body forces, and incompressibility separately. Our model only involves a replacement of the usual Poisson equation and velocity update in the pressure projection step which solves for incompressibility. This is illustrated in figure 1 which shows one iteration of our guiding framework: The low-resolution guiding velocity field \( \mathbf{v}_g \) is obtained using a traditional simulation pipeline (in purple) or through art direction. The guiding weights \( \phi \) are then computed as explained in section 5. Subsequently, the guided high-resolution velocity field is advected and body forces are added to obtain an intermediate velocity field \( \mathbf{v}_f \). Finally, our modified pressure projection step uses \( \Phi \) and an upsampled version of \( \mathbf{v}_g \) to obtain the new high-resolution guided velocity field \( \mathbf{v} \). To sum up, the contributions of this paper are:

- A novel mathematical model that leads to more efficient time-dependent guiding of smoke animations.
- Exploration of time-dependent guiding for artistic effects and for increasing the amount of high frequency features in the high-resolution guided flows.

An inherent assumption in our work is that the bulk movement of the flow can be satisfactorily represented at low resolution. Furthermore, our method cannot be used to specif-

![Figure 1: One simulation iteration using our method. A traditional fluid simulation pipeline (in purple) is equipped with a modified pressure projection step. Components which constitute our contributions are highlighted in green.](Image 335x630 to 442x716)

2. Related Work

Controlling Navier-Stokes based simulations at a higher level than parameter tweaking is an idea that was first introduced by Foster and Metaxas [FM97]. More recently Treuille et al. [TMPS03] proposed a key-frame control framework for smoke simulations which uses a gradient descent based optimizer. In particular they optimize for forces which make the smoke assume key-framed poses. Later McNamara et al. [MTPS04] made this approach faster by employing the adjoint method.

Shi et al. [SY05] also consider key-frame control but focus on liquid simulations. They suggest a force-based solution which allows the liquid to follow rapidly changing target shapes. Other approaches similar to Shi et al. have considered the problem of making smoke follow target shapes [FL04, HK04]. Finally, Kim et al. [KMT06] propose a method for constructing path-guided smoke trails.

Guiding of liquid simulations based on the Lattice Boltzmann Method and Smoothed Particle Hydrodynamics has been demonstrated by Thürey et al. [TKPR06]. Their method is based on controlling the low frequencies of simulations through forces which are applied near Lagrangian control particles. Nielsen et al. [NCZ’09] show that applying the idea of blending low frequency components of a simulation with a guiding velocity field before the pressure projection step may lead to noise developing over time for Eulerian smoke simulations. To avoid such artifacts they take the approach of formulating guiding as a constrained minimization problem. Concurrently, Mullen et al. [MCP’09] developed energy preserving integrators, which can also be employed to achieve higher correspondence between simulations at low and high resolution.

Contrary to the above guiding approaches where the high frequency detail arises from a physically based simulation, a lot of work has recently focused on procedural methods that add high frequency detail to a low-resolution input simula-
tion [SB08,KTJG08,NSCL08,PTSG09]. Common to these methods is that they improve both the run times and storage requirements over a full fluid simulation. The goal of Narain et al. [NSCL08] and Pfaff et al. [PTSG09] is essentially different from ours in that the synthesized flow should resemble the high-resolution flow, as opposed to the low-resolution flow, as well as possible. Narain et al. do note that the coupling from high frequencies to low frequencies can be disabled in order to retain the frequencies of the low-resolution flow, but do not focus on this scenario. On the other hand, Schechter and Bridson [SB08] and Kim et al. [KTJG08] retain the low frequency flow and procedurally add high frequency detail as a post-process. As noted by [SB08] stable, laminar structures can be forced turbulent, a property shared by [KTJG08], whereas Narain et al. appear to have solved this problem. Of the above methods, only Pfaff et al. consider turbulent structures emanating from interaction with boundaries at high resolution. However, their method does not guarantee correspondence between the low- and high-resolution flows. Our proposed guiding method inherits the ability from the underlying physical simulation to automatically develop instabilities, including those arising in the vicinity of boundaries.

### 3. Guiding

In this paper we define **guiding** as follows: An input velocity field, \( v_g \), is used to enforce a correspondence between the low frequencies of a higher-resolution, simulated velocity field, \( v \), and \( v_g \) itself. The input velocity field, \( v_g \), can originate from a physically based simulation or be constructed by an artist by any other means such as illustrated in figure 5. The goal of this section is to formulate guiding as a constrained minimization problem. The minimization will enforce the guiding, whereas the **constraint** will enforce that the resulting velocity field is divergence free. By employing calculus of variations and the method of Lagrange multipliers, the minimization problem is transformed into a set of partial differential equations, the Euler-Lagrange equations, that solve for the corresponding stationary point. In this paper we will denote this set of partial differential equations the **guiding equations**. Note that the Poisson equation solving for incompressibility in a standard fluid solver can in fact also be derived from a constrained minimization problem [FP02, pp. 202-204].

Ideally, the guiding should only affect the low frequencies of \( v \). Therefore we separate the low and high frequencies in our mathematical model. In addition, this separation leads to guiding equations where the left-hand-side, *i.e.* the matrix, is independent of the spatially and temporally varying guiding weights which we denote \( \phi(x,t) \in [0;1] \) is a time-dependent scalar field that enforces the strength of the guiding. If \( \phi = 1 \) in a given point, guiding is enforced with maximal strength (see second image from the left in figure 3). If on the other hand \( \phi = 0 \), the guiding equations essentially reduce to the normal Poisson equation and velocity update, hence no guiding is enforced. The scalar field \( \phi \) is computed as described in section 5 prior to solving the guiding equations for pressure and velocity.

We first state our mathematical model and then elaborate on an intuitive interpretation below. In particular we wish to minimize

\[
\int_{\Omega} (1 - \phi(x)) \| [F * v] - \tilde{v}(x) \| (x) \| F * v(x) \| (x) \| v - [F * v] \| (x) \| x \right) \| d \Omega \right) + \int_{\Omega} \| v - [F * v] \| (x) \| v - [F * v] \| (x) \| x \right) \| d \Omega \right)
\]

subject to the constraint

\[\nabla \cdot v(x) = 0\]

where \( \Omega \) is the fluid simulation domain, \( \tilde{v} \) is the current high resolution velocity field in the operator splitting sequence (*i.e.* the velocity field from the previous frame updated by self-advection and forces), \( F \) is a lowpass filter, and \( * \) is the convolution operator. Contrary to the approach taken in [NCZ09,SB08,], we separate the low and high frequencies in the minimization problem. Specifically, expressions (1) and (2) of the minimization problem involve the low frequencies of \( v \), expressed as the convolution \( [F * v] \), whereas (3) involves the remaining high frequencies, \( v - [F * v] \), of \( v \). Expression (1) (expression (2)) contains the low frequencies of the difference between the current high resolution velocity field (the guiding velocity field) and the velocity field we are solving for. Expression (3) on the other hand contains the high frequencies of the difference between the current high resolution velocity field and the velocity field we are solving for. By keeping (3) independent of the guiding weights it turns out that the resulting matrix also becomes independent of the guiding weights. Finally, Eq. (4) is the constraint that forces the resulting velocity field to be divergence free.

If \( \phi = 0 \), (2) vanishes, and the minimization problem reduces to a minimization of the difference between \( v \) and \( \tilde{v} \). Furthermore, if \( \phi = 0 \) and \( F \) is a perfect lowpass filter, (1) and (3) can be combined into \( \int_{\Omega} (v(x) - \tilde{v}(x)) \| d \Omega \right) \| x \right) \| which can be seen by transforming to Fourier space and applying Parseval's theorem. Minimizing \( \int_{\Omega} (v(x) - \tilde{v}(x)) \| d \Omega \right) \| subject to the constraint Eq. (4) is in fact equivalent to solving the usual Poisson equation and subsequent velocity update [FP02]. Hence, the guiding equations will essentially result in an unguided simulation for \( \phi = 0 \).

If \( \phi = 1 \), the low frequencies of the resulting velocity field \( v \) will minimize the difference to \( v_g \) alone and hence only the guiding will be in effect, see (2). On the other hand \( \phi \in (0;1) \) will minimize the difference to both \( v_g \) and the low frequencies of \( \tilde{v} \), and thus \( v \) will be a blend of the two low-frequency components. It is an important point of this paper that relaxing the strength of the guiding by lowering \( \phi \) whenever possible is key to allowing more turbulence to develop naturally in guided simulations.
To turn our proposed minimization problem into the corresponding Euler-Lagrange equations (the guiding equations) we employ calculus of variations. The full derivation is included in appendix A. Here we merely present the result and explain its consequences. Specifically, the constrained minimization problem (1) – (4) leads to the following equations:

\[ \psi(x) + 2 \int_{\Omega} \mathcal{F}(y - x) |[\mathcal{F} \ast v](y) - v(y)| dy + \nabla p(x) = 0 \]

\[ \nabla \cdot v(x) = 0 \]  

(5)

(6)

where \( p \) is the pressure of the fluid. There is one equation for each point \( x \) in the fluid simulation domain. All equations are solved simultaneously for both the guided velocity field, \( v \), as well as the pressure, \( p \). On a staggered grid, Eq. (5) leads to one equation for each face, and Eq. (6) leads to one equation for each cell center. This amounts to \((D + 1)N\) equations in \((D + 1)N\) unknowns, where \( N \) is the number of grid points. Note that a standard fluid solver also has to solve for \((D + 1)N\) unknowns, velocity and pressure. However, in this case there is only one a two-way coupling between the unknown pressure and velocity and hence the Poisson equation involves only the pressure, \( N \) equations in \( N \) unknowns. The derived equations are amenable to solution in the form stated above. However, for solution using relaxation methods such as Jacobi or Gauss-Seidel (including the multigrid method which utilizes these relaxation methods) it is convenient to replace Eq. (6) by Eq. (5) with the divergence operator applied to both sides of the equation. By exploiting the constraint, Eq. (6), \( \nabla \cdot v(x) \) vanishes from the resulting equation and hence the constraint remains enforced. Furthermore, this will ensure non-zero diagonal entries as required by the relaxation methods. Eq. (5) and Eq. (6) share the properties of the model of [NCZ'09] that if \( v = \mathcal{F} \ast v \) and \( v \) is divergence free, the resulting velocity field will be \( v \) itself. Also, \( v \) does not have to be divergence free. In addition, our new model has the following properties not shared by [NCZ’09]:

- The guiding weights, \( \phi \), appear only on the right-hand-side of the equations. This means that whenever \( \phi \) depends on time, the matrix operators do not have to be recomputed.
- Spatially varying \( \phi \) will not increase the storage requirements of the matrix operators since \( \phi \) appears only on the right-hand-side, hereby also facilitating faster computation of the matrix operators.

4. Implementation

The guiding equations are discretized on a staggered grid with velocities sampled on cell faces and pressure sampled in cell centers [FS10]. We use BFECC [DL07] and monotonic, cubic interpolation [FS10] for advection. To solve the equations we employ a parallel multigrid implementation [BHM00] combined with a simple compression scheme of the matrix operators similar to the one described in [NCZ’09], except that we use a relaxation parameter of 0.9 in our SOR implementation. Boundaries are handled by applying the penalization method of Angot et al. [ABF99, KCR08]. Finally, upsampling and lowpass filter estimation is handled as in [NCZ’09].

5. Time-Dependent Guiding Effects and Results

Guiding is applied to enforce correspondence between the low- and high-resolution velocity fields. However, although our guiding model is separating the low and high frequencies, it is evident in practice that imposing too much guiding will constrain the high frequencies as well. On the other hand, relaxing the guiding weights uniformly will cause the guided simulation to behave increasingly as an unguided simulation, thereby developing more turbulent structures. However, this may come at the cost of losing correspondence to the low-resolution velocity field, see figure 3.

In the following we explore a number of different approaches to guiding the high-resolution simulation with both spatially and temporally dependent guiding weights, \( \phi(x,t) \), e.g. to allow more turbulent structures to emerge. The guiding weights are represented as a time-dependent scalar field fed into the linear equation solver in each iteration. It is important to note that the high frequency features that develop in the guided simulations are instabilities arising in the underlying high-resolution, physically based simulation. Vorticity confinement or procedural turbulence can of course be added on top of the guided simulations to synthesize an even more turbulent behaviour.

5.1. Time-Dependent Guiding with Smoke Density

Using the smoke density as a guiding weight is based on the observation that in many cases we want thick volumes of smoke to adhere to the bulk motion prescribed by the low-resolution guiding velocity field. On the other hand we also want turbulence to develop in areas where the
Figure 3: Identical frames from several simulations of a smoke column at resolution $64 \times 256 \times 64$ rising due to a buoyant force. The leftmost image depicts a low resolution simulation and the rightmost image depicts a high resolution simulation. The remaining images from left to right: (1) Our method with a uniform weight of $\phi = 1$. (2) Our method with a uniform weight of $\phi = 0.02$. The amount of turbulence along the column is similar to the non-uniform guiding in the image to the right. However, using uniform weights, the column rises faster than the low-resolution simulation. (3) Our method with eroded densities combined with the error estimate, $\phi_{low} = 0$ and $\phi_{high} = 0.35$. (4) Same approach with $\phi_{low} = 0$ and $\phi_{high} = 0.55$. (5) Same approach with $\phi_{low} = 0.15$ and $\phi_{high} = 0.35$. Notice that the relatively high $\phi_{low}$ results in a lack of high frequency detail.

Figure 4: Left: A rising smoke column interacting with a translating Stanford bunny at resolution $128 \times 256 \times 64$. Upper row: Early frame. From left to right: The low resolution guiding simulation, the guiding approach of [NCZ*09], our guiding with eroded densities, our guiding with eroded densities combined with the error estimate and finally the high resolution simulation. Lower row: Later frame. Right: Two guided simulations of a Stanford bunny rotating close to a smoke jet subject to a buoyant force at resolution $128^3$. Top: The guiding method of [NCZ*09]. Bottom: Our method guiding with eroded densities. The addition of turbulent features is more subtle and less noticeable in this case, but is visible for example in the upper left corner.

Smoke is wispy. Let the smoke density be denoted by $d(x,t)$ and assume that $d(x,t) \in [0;1]$. We construct the guiding weights as $\phi(x,t+\Delta t) = \phi_{high}d(x,t) + \phi_{low}(1-d(x,t))$, where $\phi_{low}$ and $\phi_{high}$ are lower and upper bounds on the guiding weights, respectively. Unless stated otherwise we use $\phi_{low} = 0$ and $\phi_{high} = 0.35$ as found by experimentation, see figure 3. In our experience, guiding with density yields better results when combined with the extensions described below.

5.1.1. Combining with Erosion of Densities

We have explored performing an erosion of the smoke density used to construct the guiding weights. The idea behind this is that the guiding will be relaxed in a wider band in regions between smoke and clear air, thereby allowing more turbulent structures to emanate in these areas. Erosion is characterized by a single parameter which is the width of a stencil centered at each grid point. The result of the erosion operation at a given grid point is the minimum density value found under the support of the stencil when centered at that particular grid point. We have found by experimentation that using a stencil width of three generates the best results. Using wider stencils tends to make the simulations behave more like unguided simulations, e.g. making smoke rise faster. Figure 2 shows an example of a simulation using the eroded density as guiding weight and compares it to uniform guiding weights throughout the domain using the model of Nielsen et al. [NCZ*09]. In all comparisons...
to [NCZ’09], we set their scaling parameter to $\alpha = 0.65$ as suggested in their paper. As is evident from the figure, more turbulence appears when the guiding weights are relaxed using our method. Additional examples are shown in figure 4.

Time-dependent guiding using [NCZ’09] in the example of the rising smoke column in figure 2 requires roughly 1600 seconds per frame to compute the matrix which takes up roughly 1.6GB of storage. Using our guiding model the matrix computation takes roughly 120 seconds, requires 74MB, and only has to be performed once for an entire simulation. The guided smoke column simulation in figure 2 runs for 36 – 37 seconds per frame using our method, depending on which approach is utilized to compute the guiding weights. In contrast an unguided simulation at the same resolution takes roughly 17 seconds per frame. A similar result was reported in [NCZ’09]. However, we emphasize that their results were obtained using steady guiding in which the guiding weights do not vary in time. If in fact the guiding weights vary in time, their method requires 1600 seconds per frame for the matrix computation plus the simulation time. In our fluid solver implementation we employ the relatively expensive combination of BFECC and cubic interpolation for high quality advection of both velocities, densities and temperature. For the unguided simulation in figure 3, these advection steps take up 46% of the total time per frame, whereas the multigrid solver uses only 22% of the time per frame. The time required to solve the guiding equations in the same example is about 5.2 times the time required to solve the Poisson equation and perform the velocity update in a standard fluid solver at the same resolution. However, since the multigrid solver only takes up 22% of the total time per frame for an unguided simulation, this explains why the total time per frame is not more than doubled, despite the fact that more unknowns are involved in our linear system of equations.

5.2. Time-Dependent Guiding with Curves

The previous examples have looked at guiding with a velocity field obtained from a low-resolution simulation. However, artistic effects can be achieved by constructing velocity fields through non-physically based methods such as modeling or painting. These can then be used as is or combined with an existing low-resolution simulation. They are applied using time-dependent guiding weights to enable artists to enhance, decrease or create new low frequency motion in an intuitive manner. We have examined this idea through the following approach: Let $\psi_{\text{curve}}(x,t)$ be an artist-specified guiding velocity field and let $\phi_{\text{curve}}(x,t)$ be similarly specified guiding weights. Essentially these fields could describe any motion the artist desires, and in particular they do not have to ensure a divergence free guiding velocity field. We will specify $\phi$ such that guiding is only applied in a narrow tube around various curves $C_1, \ldots, C_n$. These same curves are used to create velocity fields $\psi_{\text{curve}_1}, \ldots, \psi_{\text{curve}_n}$ which flow along them. $\phi$ is animated over time by gently increasing and decreasing local guiding weights $\phi_{\text{curve}_1}, \ldots, \phi_{\text{curve}_n}$ in order to gradually enable and disable the flows induced by the corresponding curves. These local guiding weights form tubes in which velocities along the curves are induced and they are illustrated in red in the leftmost pair of images of figure 5. In the animation we consider, the two guiding curves are enabled one after the other. As stated, these guiding effects can also be combined with error estimates and smoke densities to produce more high frequency detail. An example of this, where these additional effects are combined with the $\psi_{\text{curve}}$, can be seen to the right in figure 5. The matrix was computed in 181 seconds and required 82MB of storage at simulation resolution 128$^3$. Each frame required 65 seconds to compute. Employing the guiding model of Nielsen et al. in this example required roughly 1800 seconds per frame to compute the matrix and roughly 3GB to store it. Note again, that the matrix computation is only required once per simulation in our framework.

5.3. Discussion and Limitations

The computation time required for a guided high-resolution simulation is roughly doubled when compared to an unguided simulation at the same resolution. Procedural methods are faster and require less storage than a standard fluid solver. However, as opposed to our guiding approach, procedural methods that ensure a correspondence between low and high resolution can force otherwise stable laminar flow to become turbulent and do not capture the instabilities naturally occurring near boundaries. In the accompanying video we compare our guiding method to the wavelet turbulence method [KTJG08] to demonstrate these differences.

Guiding with the error estimate does not always result in significantly higher detail, and for some simulations the similarity between low and high resolution can become too weak. Another potential problem is that all grid points in
the domain are coupled by the error estimate. This may be improved by using absolute errors combined with clamping and/or nonlinear interpolation. Furthermore, incorporation of the density field and error into the guiding weights does not currently ensure density matching between the low- and high-resolution simulations. In the future we wish to explore density matching which may be used to ensure similarity for guiding with the error estimate and ensure that no drift occurs over time. Note that [NCZ'09] also does not prevent drift of densities, since their variational model considers only velocities. Additionally, exploring other guiding weights e.g. based on the gradient of density as opposed to density alone is an interesting direction for future work.

Finally, for some simulations a simpler method such as blending the low frequencies of the high resolution velocity field with the guiding velocity field might suffice [TKPR06], provided that artifacts do not appear within the time-frame required by the simulation.

6. Conclusion
Smoke animations are hard to art-direct because of numerical viscosity and the non-linear nature of the governing in-viscid Euler equations. In this paper we propose an improved variational method for guiding high-resolution smoke simulations using low-resolution velocity fields. The method is faster and requires less storage for time-dependent guiding than previous methods. We have demonstrated that time-dependent guiding can be used to achieve more high frequency detail and achieve artistic effects. In conclusion, we are convinced that the variational approach adopted in our method can be applied to a large number of fluid control paradigms. We also believe that further exploration into determining the guiding weights φ can yield new ways of creating artistic effects.

Acknowledgements
This work was partially funded by the Danish Agency for Science, Technology and Innovation. We wish to thank Petter Trier for acting as a consultant on rendering issues (see http://cg.alexandria.dk/), and Jesper Mosegaard, Thomas Sangild, Clemens Nylandsted Klokmos, Ole Østerby and the anonymous reviewers for their comments. We are grateful to Ken Museth, Doug Roble, Nafees Bin Zafar, Michael Clive and Ryo Sakaguchi for inspiring discussions.

References


Expanding $Q_{\text{min}}$ gives

$$Q_{\text{min}} = \int_{\Omega} \{ \mathbf{v} \cdot \mathbf{v} + |\mathbf{F} \ast \mathbf{v}| \cdot |\mathbf{F} \ast \mathbf{v}| + \mathbf{C} : \mathbf{C} - 2|\mathbf{F} \ast \mathbf{v}| \cdot \mathbf{v} - 2|\mathbf{F} \ast \mathbf{v}| \cdot \mathbf{C} + 2(\mathbf{v} + \partial \mathbf{v}) \cdot \mathbf{C} \} \, d\mathbf{x}$$

Expanding $Q(\mathbf{v} + \partial \mathbf{v})$ further and omitting higher order terms in $\partial \mathbf{v}$ such as $2\partial \mathbf{v} \cdot (\mathbf{F} \ast \partial \mathbf{v})$, $\partial \mathbf{v} \cdot \partial \mathbf{v}$ and $|\mathbf{F} \ast \partial \mathbf{v}| : |\mathbf{F} \ast \partial \mathbf{v}|$, for $\delta Q = Q(\mathbf{v} + \partial \mathbf{v}) - Q_{\text{min}}$ we obtain (to first order in $\partial \mathbf{v}$)

$$\delta Q = 2 \int_{\Omega} \{ \mathbf{v} \cdot \mathbf{v} + |\mathbf{F} \ast \mathbf{v}| \cdot |\mathbf{F} \ast \mathbf{v}| + \mathbf{C} : \mathbf{C} - 2|\mathbf{F} \ast \mathbf{v}| \cdot \mathbf{v} - 2|\mathbf{F} \ast \mathbf{v}| \cdot \mathbf{C} + 2(\mathbf{v} + \partial \mathbf{v}) \cdot \mathbf{C} \} \, d\mathbf{x}$$

To obtain a sufficient condition for a stationary point it is convenient to rewrite $\delta Q$ in the form $\delta Q = \int_{\Omega} \delta \mathbf{v} \cdot (\mathbf{D} \cdot \mathbf{x}) \, d\mathbf{x}$, where $\mathbf{D}$ is some vector expression. As can be seen, (9) is already in this form, and we now focus on rewriting (10) in this form as well. By utilizing the definition of the convolution operator, exchanging integration orders and introducing the integration variable $\mathbf{y}$ which ranges over all positions in the domain, we obtain from (9) and (10)

$$\delta Q = 2 \int_{\Omega} \{ \mathbf{v} \cdot \mathbf{v} + |\mathbf{F} \ast \mathbf{v}| \cdot |\mathbf{F} \ast \mathbf{v}| + \mathbf{C} : \mathbf{C} - 2|\mathbf{F} \ast \mathbf{v}| \cdot \mathbf{v} - 2|\mathbf{F} \ast \mathbf{v}| \cdot \mathbf{C} + 2(\mathbf{v} + \partial \mathbf{v}) \cdot \mathbf{C} \} \, d\mathbf{x}$$

Since $\partial \mathbf{v}$ is arbitrary, a sufficient condition for $\delta Q = 0$ is that the expression within the curly braces is identically zero. This completes the derivation of the contribution to the Euler-Lagrange equations from (3). The contributions from (1) and (2) are derived similarly. Furthermore, the derivation of the contribution, $\nabla \rho(\mathbf{x})$, from Eq. (4) by the method of Lagrange multipliers is similar to the derivation in [FP02, p.203]. The contributions from (1) – (4) are finally added together and set equal to zero

$$\int_{\Omega} \{ 1 - \phi(y) |\mathbf{F} \ast \mathbf{v}| (\mathbf{F} \ast \mathbf{v})(\mathbf{y}) - (\mathbf{F} \ast \mathbf{v})(\mathbf{y}) \} \, d\mathbf{y}$$

Expression (11) is the contribution from (1), (12) is the contribution from (2), (13) – (14) are the contributions from (3) and finally (15) is the contribution from Eq. (4). Note that $\rho$ is the pressure of the fluid. By re-arranging these equations and moving the known quantities to the right-hand-side we obtain Eq. (5). This completes the derivation.