summary of "Visual Simulation of Smoke"

2019 年 5 月 13 日

1 abstract

This paper propose an unconditionally stable method of inviscid, incompressible, constant density fluid simulation. A main feature of this method is using the technique "Vorticity Confinement". This technique compensate energy loss of numerical dissipation.

2 Principle

2.1 Equation of motion

We will consider dynamics of inviscid, incompressible, constant density fluids. A typical example is smoke. Equation of motion is described by Euler equation

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \mathbf{f} \tag{1}$$

and incompressible condition

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

where $\mathbf{u}$ is velocity field, p is pressure and $\mathbf{f}$ is external force. Each term of Euler equation represent following physical meanings:

- 1st term of right hand side: advection
- 2nd term of right hand side: the effect of pressure
- 3rd term of right hand side: external force

The external force term consists of buoyancy force term and vorticity confinement term.

$$\mathbf{f} = \mathbf{f}_{buoy} + \mathbf{f}_{conf}, \tag{3}$$

$$\mathbf{f}_{buoy} := -\alpha\rho\mathbf{z} + \beta(T - T_{amb})\mathbf{z} \tag{4}$$

where $\mathbf{z}$ is vertical direction vector, T_{amb} is ambient air temperature and α, β are constant.Each term of buoyancy force represent gravitational force and Fourier's law. Vorticity confinement term is represented as eq. (13)

Fluids cast smoke's density and temperature along with its dynamics. We assume that Equation of motion of smoke's density and temperature is following advect equations:

$$\frac{\partial \rho}{\partial t} = -\mathbf{u} \cdot \nabla\rho \tag{5}$$

$$\frac{\partial T}{\partial t} = -\mathbf{u} \cdot \nabla T \tag{6}$$

where ρ is smoke's density and T is temperature field.

calculation of pressure gradient term

Euler eq. (1) contains unknown variable p, so we will explain how to calculate pressure gradient term. Calculation of pressure gradient term is realized by solving Poisson equation. First, we evaluate intermediate velocity $\mathbf{u}^*$ by using right hand side of the Euler eq. without pressure gradient term

$$\mathbf{u}^* = \mathbf{u}(t) + \Delta t(-(\mathbf{u}(t) \cdot \nabla)\mathbf{u}(t) + \mathbf{f}). \tag{7}$$

where advection term is calculated by the method of characteristics that holds numerical stability. Next, we consider time evolution by pressure gradient term

$$\frac{\mathbf{u}(t + \Delta t) - \mathbf{u}^*}{\Delta t} = -\frac{1}{\rho}\nabla p. \tag{8}$$

We impose the incompressible condition$\nabla \cdot \mathbf{u}(t + \Delta t) = 0$ to this equation and we have Poisson eq.

$$\nabla^2 p = \frac{1}{\Delta t}\nabla \cdot \mathbf{u}^*. \tag{9}$$

This equation is discretized as sparse linear equation and then get pressure field by solving it. By taking gradient of pressure field, we get the velocity after one time step

$$\mathbf{u}(t + \Delta t) = \mathbf{u}^* - \Delta t \nabla \cdot p \tag{10}$$

Vorticity Confinement

We will explain the vorticity confinement term in external force eq. (3). As stated above, advection term is calculated by method of characteristics. This method has unconditionally stable property because maximum velocity after the method is equals or smaller than maximum velocity before the method. But this method does not guarantee energy conservation(numerical dissipation). Vorticity confinement term in the external force compensate this energy loss.

We define vorticity $\boldsymbol{\omega}$ by

$$\boldsymbol{\omega} := \nabla \times \mathbf{u} \tag{11}$$

and gradient of vorticity amplitude $\mathbf{N}$

$$\mathbf{N} := \frac{\boldsymbol{\eta}}{|\boldsymbol{\eta}|} \qquad (\boldsymbol{\eta} := \nabla|\omega|), \tag{12}$$

then vorticity confinement term $\mathbf{f}_{conf}$ is given by

$$\mathbf{f}_{conf} = \epsilon h(\mathbf{N} \times \boldsymbol{\omega}) \tag{13}$$

where h is length of grid cell and $\epsilon > 0$ is controllable parameter.

This term strengthen the effect of vortices and reduce loss of information about small scale due to numerical dissipation(see Fig. 1).

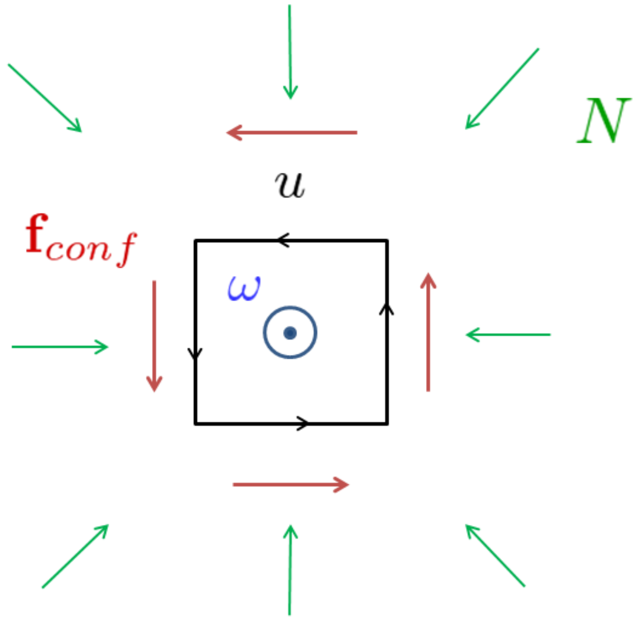

Fig. 1 Physical meaning of vorticity confinement. We consider one vortex which have vorticity perpendicular to the paper. We assume that vorticity amplitude far from the vortex is small and **N** direct to the vortex. In this situation, the direction of vorticity confinement term is parallel to velocity field and strengthen vorticity.

2.2 discretization of Equation of motion

We will solve eq. (1) by discretizing in temporal and spatial way. Time evolution from t to $t + \Delta t$ is calculated by following 3 steps:

1. add force
2. advect
3. calculate pressure gradient term

$$\mathbf{w}_0(\mathbf{x})(:= \mathbf{u}(\mathbf{x}, t)) \xrightarrow{addforce} \mathbf{w}_1(\mathbf{x}) \xrightarrow{advect} \mathbf{w}_2(\mathbf{x}) \xrightarrow{calc\ pressure\ gradient} \mathbf{w}_3(\mathbf{x})(:= u(\mathbf{x}, t + \Delta t)). \tag{14}$$

In each step, temperature, smoke's density, external forces are defined in the centers of voxels and velocities are defined at the cell faces. We will denote the temperature, smoke's density, external forces at center of voxel(i, j, k) as

$$T_{(i,j,k)},\ \rho_{(i,j,k)},\ f_{(i,j,k)}, \qquad i, j, k = 1, \ldots N \tag{15}$$

and velocities as

$$u_{x,(i+1/2,j,k)}, \qquad i = 0, \ldots N, \quad j, k = 1, \ldots N \tag{16}$$

$$u_{y,(i,j+1/2,k)}, \qquad j = 0, \ldots N, \quad i, k = 1, \ldots N \tag{17}$$

$$u_{z,(i,j,k+1/2)}, \qquad k = 0, \ldots N, \quad i, j = 1, \ldots N \tag{18}$$

where N is number of grid cells in one direction. $\mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3$ are denoted in the same manner as $\mathbf{u}$.

We will explain each step of time evolution.

1. add force

In this step, we calculate external force term

$$w_1(\mathbf{x}) = w_0(\mathbf{x}) + \Delta t \mathbf{f}(\mathbf{x}, t). \tag{19}$$

In each grids, this equation is

$$w_{1,x,(i+1/2,j,k)} = w_{0,x,(i+1/2,j,k)} + \Delta t f(t)_{x,(i+1/2,j,k)} \tag{20}$$
$$w_{1,y,(i,j+1/2,k)} = w_{0,x,(i,j+1/2,k)} + \Delta t f(t)_{y,(i,j+1/2,k)} \tag{21}$$
$$w_{1,z,(i,j,k+1/2)} = w_{0,x,(i,j,k+1/2)} + \Delta t f(t)_{z,(i,j,k+1/2)} \tag{22}$$

where $f(t)_{x,(i+1/2,j,k)},\ f(t)_{y,(i,j+1/2,k)},\ f(t)_{z,(i,j,k+1/2)}$ are averages of two adjacent faces' value.

$$f(t)_{x,(i+1/2,j,k)} = (f(t)_{x,(i,j,k)} + f(t)_{x,(i+1,j,k)})/2 \tag{23}$$
$$f(t)_{y,(i,j+1/2,k)} = (f(t)_{y,(i,j,k)} + f(t)_{y,(i,j+1,k)})/2 \tag{24}$$
$$f(t)_{z,(i,j,k+1/2)} = (f(t)_{z,(i,j,k)} + f(t)_{z,(i,j,k+1)})/2. \tag{25}$$

We derive the expression of discretized vorticity confinement term. Vorticity $\boldsymbol{\omega}_{(i,j,k)}$ is expressed as

$$\omega_{x,(i,j,k)} = (u_{z,(i,j+1,k)} - u_{z,(i,j-1,k)} - u_{y,(i,j,k+1)} + u_{y,(i,j,k-1)})/2h \tag{26}$$
$$\omega_{y,(i,j,k)} = (u_{x,(i,j,k+1)} - u_{x,(i,j,k-1)} - u_{z,(i+1,j,k)} + u_{z,(i-1,j,k)})/2h \tag{27}$$
$$\omega_{z,(i,j,k)} = (u_{y,(i+1,j,k)} - u_{y,(i-1,j,k)} - u_{x,(i,j+1,k)} + u_{x,(i,j-1,k)})/2h \tag{28}$$

where $u_{x,(i,j,k)}, u_{y,(i,j,k)}, u_{z,(i,j,k)}$ are cell centered velocity

$$u_{x,(i,j,k)} = (u_{x,(i+1/2,j,k)} + u_{x,(i-1/2,j,k)})/2 \tag{29}$$
$$u_{y,(i,j,k)} = (u_{y,(i,j+1/2,k)} + u_{y,(i,j+1/2,k)})/2 \tag{30}$$
$$u_{z,(i,j,k)} = (u_{z,(i,j,k+1/2)} + u_{z,(i,j,k+1/2)})/2. \tag{31}$$

Then we can get $\eta = \nabla|\boldsymbol{\omega}|$

$$\nabla|\boldsymbol{\omega}| = \begin{pmatrix} \left(\frac{\partial|\omega|}{\partial x}\right)_{x,(i+1/2,j,k)} \\ \left(\frac{\partial|\omega|}{\partial y}\right)_{x,(i,j+1/2,k)} \\ \left(\frac{\partial|\omega|}{\partial z}\right)_{x,(i,j,k+1/2)} \end{pmatrix} = \begin{pmatrix} (|\omega|_{(i+1,j,k)} - |\omega|_{(i,j,k)})/h \\ (|\omega|_{(i,j+1,k)} - |\omega|_{(i,j,k)})/h \\ (|\omega|_{(i,j,k+1)} - |\omega|_{(i,j,k)})/h \end{pmatrix} \tag{32}$$

and cell centered η is

$$\boldsymbol{\eta}_{(i,j,k)} = \begin{pmatrix} \eta_{x,(i,j,k)} \\ \eta_{y,(i,j,k)} \\ \eta_{z,(i,j,k)} \end{pmatrix} = \begin{pmatrix} (\eta_{x,(i+1/2,j,k)} + \eta_{x,(i-1/2,j,k)})/2 \\ (\eta_{y,(i,j+1/2,k)} + \eta_{y,(i,j+1/2,k)})/2 \\ (\eta_{z,(i,j,k+1/2)} + \eta_{z,(i,j,k+1/2)})/2 \end{pmatrix}. \tag{33}$$

So we can get discretized vorticity confinement term

$$\mathbf{f}_{conf,(i,j,k)} = \frac{\epsilon h}{|\boldsymbol{\eta}_{(i,j,k)}|} \begin{pmatrix} \eta_{y,(i,j,k)}\omega_{z,(i,j,k)} - \eta_{z,(i,j,k)}\omega_{y,(i,j,k)} \\ \eta_{z,(i,j,k)}\omega_{x,(i,j,k)} - \eta_{x,(i,j,k)}\omega_{z,(i,j,k)} \\ \eta_{x,(i,j,k)}\omega_{y,(i,j,k)} - \eta_{y,(i,j,k)}\omega_{x,(i,j,k)} \end{pmatrix}. \tag{34}$$

2. advect

In this step, we calculate advect term. we use method of characteristics for updating velocity field(see Appendix A)

$$w_{2,x}(x,y,z) = w_{1,x}(x - u_x\Delta t, y - u_y\Delta t, z - u_z\Delta t) \tag{35}$$
$$w_{2,y}(x,y,z) = w_{1,y}(x - u_x\Delta t, y - u_y\Delta t, z - u_z\Delta t) \tag{36}$$
$$w_{2,z}(x,y,z) = w_{1,z}(x - u_x\Delta t, y - u_y\Delta t, z - u_z\Delta t) \tag{37}$$

where $w_{2,\alpha}(\alpha = x, y, z)$ is α component of $\mathbf{w}_2$. I each grids

$$w_{2,x,(i+1/2,j,k)} = w_{1,x,(i+1/2,j,k)}(x - u_{x,(i+1/2,j,k)}\Delta t, y - u_{y,(i+1/2,j,k)}\Delta t, z - u_{z,(i+1/2,j,k)}\Delta t) \tag{38}$$
$$w_{2,y,(i,j+1/2,k)} = w_{1,y,(i,j+1/2,k)}(x - u_{x,(i,j+1/2,k)}\Delta t, y - u_{y,(i,j+1/2,k)}\Delta t, z - u_{z,(i,j+1/2,k)}\Delta t) \tag{39}$$
$$w_{2,z,(i,j,k+1/2)} = w_{1,z,(i,j,k+1/2)}(x - u_{x,(i,j,k+1/2)}\Delta t, y - u_{y,(i,j,k+1/2)}\Delta t, z - u_{z,(i,j,k+1/2)}\Delta t) \tag{40}$$

where $u_{y,(i+1/2,j,k)}, u_{z,(i+1/2,j,k)}, u_{x,(i,j+1/2,k)}, u_{z,(i,j+1/2,k)}, u_{x,(i,j,k+1/2)}, u_{y,(i,j,k+1/2)}$ are given by four points averages

$$u_{y,(i+1/2,j,k)} = (u_{y,(i,j+1/2,k)} + u_{y,(i,j-1/2,k)} + u_{y,(i+1,j+1/2,k)} + u_{y,(i+1,j-1/2,k)})/4 \tag{41}$$
$$u_{z,(i+1/2,j,k)} = (u_{z,(i,j,k+1/2)} + u_{z,(i,j,k-1/2)} + u_{z,(i+1,j,k+1/2)} + u_{z,(i+1,j,k-1/2)})/4 \tag{42}$$
$$u_{x,(i,j+1/2,k)} = (u_{x,(i+1/2,j,k)} + u_{z,(i-1/2,j,k)} + u_{z,(i+1/2,j+1,k)} + u_{z,(i-1/2,j+1,k)})/4 \tag{43}$$
$$u_{z,(i,j+1/2,k)} = (u_{z,(i,j,k+1/2)} + u_{z,(i,j,k-1/2)} + u_{z,(i,j+1,k+1/2)} + u_{z,(i,j+1,k-1/2)})/4 \tag{44}$$
$$u_{x,(i,j,k+1/2)} = (u_{x,(i+1/2,j,k)} + u_{z,(i-1/2,j,k)} + u_{z,(i+1/2,j,k+1)} + u_{z,(i-1/2,j,k+1)})/4 \tag{45}$$
$$u_{y,(i,j,k+1/2)} = (u_{y,(i,j+1/2,k)} + u_{y,(i,j-1/2,k)} + u_{y,(i,j+1/2,k+1)} + u_{y,(i,j-1/2,k+1)})/4. \tag{46}$$
$$\tag{47}$$

In general, end points of trace path $(x - u_x\Delta t, y - u_y\Delta t, z - u_z\Delta t)$ don't confirm with a face of voxels. So we estimate velocity after advection by Monotonic Cubic Interpolation. (see Appendix. B)

3. calculate pressure gradient term

In this step, we solve Poisson equation and calculate pressure gradient term. First, we evaluate pressure field by solving Poisson equation

$$\nabla^2 p = \frac{1}{\Delta t}\nabla \cdot \mathbf{w}_2 \tag{48}$$

Laplacian of pressure is discretized as followings

$$(\nabla^2 p)_{i,j,k} = \frac{p_{i+1,j,k} + p_{i-1,j,k} + p_{i,j+1,k} + p_{i,j-1,k} + p_{i,j,k+1} + p_{i,j,k-1} - 6p_{i,j,k}}{h^2} \tag{49}$$

and divergence of $\mathbf{w}_2$ is discretized as

$$(\nabla \cdot w_2)_{i,j,k} = \frac{w_{2,x,(i+1/2,j,k)} - w_{2,x,(i-1/2,j,k)} + w_{2,y,(i,j+1/2,k)} - w_{2,y,(i,j-1/2,k)} + w_{2,z,(i,j,k+1/2)} - w_{2,z,(i,j,k-1/2)}}{h}. \tag{50}$$

So, we get linear equation

$$\sum_{i',j',k'} (\frac{1}{h^2}(\delta_{(i+1,j,k),(i',j',k')} + \delta_{(i-1,j,k),(i',j',k')} + \delta_{(i,j+1,k),(i',j',k')} + \delta_{(i,j-1,k),(i',j',k')} \tag{51}$$
$$+\delta_{(i,j,k+1),(i',j',k')} + \delta_{(i,j,k-1),(i',j',k')} - 6\delta_{(i,j,k),(i',j',k')}))p_{(i',j',k')} \tag{52}$$
$$= \frac{w_{2,x,(i+1/2,j,k)} - w_{2,x,(i-1/2,j,k)} + w_{2,y,(i,j+1/2,k)} - w_{2,y,(i,j-1/2,k)} + w_{2,z,(i,j,k+1/2)} - w_{2,z,(i,j,k-1/2)}}{\Delta t h} \tag{53}$$

and by solving this equation, we have pressure field p. We take gradient of pressure field and subtract from $\mathbf{w}_2$, then we finally get velocity field after one time step

$$w_{3,x,(i+1/2,j,k)} = w_{2,x,(i+1/2,j,k)} - \left(\frac{\partial p}{\partial x}\right)_{(i+1/2,j,k)} \tag{54}$$

$$w_{3,y,(i,j+1/2,k)} = w_{2,y,(i,j+1/2,k)} - \left(\frac{\partial p}{\partial y}\right)_{(i,j+1/2,k)} \tag{55}$$

$$w_{3,z,(i,j,k+1/2)} = w_{2,z,(i,j,k+1/2)} - \left(\frac{\partial p}{\partial z}\right)_{(i,j,k+1/2)} \tag{56}$$

where

$$\left(\frac{\partial p}{\partial x}\right)_{(i+1/2,j,k)} = \frac{p_{(i+1,j,k)} - p_{(i,j,k)}}{h} \tag{57}$$

$$\left(\frac{\partial p}{\partial y}\right)_{(i,j+1/2,k)} = \frac{p_{(i,j+1,k)} - p_{(i,j,k)}}{h} \tag{58}$$

$$\left(\frac{\partial p}{\partial z}\right)_{(i,j,k+1/2)} = \frac{p_{(i,j,k+1)} - p_{(i,j,k)}}{h} \tag{59}$$

Calculation method of advect term of smoke's density and temperature are same as fluids.

2.3 summary of the algorithm

The algorithm of one time step is summarized as followings.

```
1: procedure ONE TIME STEP
2:     UpdateFluidVelocityField
3:     UpdateTemperatureAndSmoke'sDensity
4: end procedure
```

Update of fluid velocity field and Moving substances are

```
1: procedure UPDATE FLUID VELOCITY FIELD
2:     addForce : eq. (20), (21), (22)
3:     advect : eq. (38), (39), (40)
4:     calculatePressureField : eq.(51)
5:     subtractPressureGradient : eq.(54), (55), (56)
6: end procedure
```

```
1: procedure UPDATE TEMPERATURE AND SMOKE'S DENSITY
2:     advectTemperature : eq.(38)
3:     advectSmoke'sDensity : eq.(38)
4: end procedure
```

Appendix A method of characteristics

method of characteristics is a technique for solving partial differential equation. We consider advect equation with initial condition

$$\frac{\partial \mathbf{u}}{\partial t}(\mathbf{x}, t) = -(\mathbf{u}(\mathbf{x}, t) \cdot \nabla)\mathbf{u}(\mathbf{x}, t) \tag{60}$$

$$\mathbf{u}(\mathbf{x}, 0) = \phi(\mathbf{x}) \tag{61}$$

We assume that $\mathbf{u}, t$ depend on a parameter s and we differentiate u with respect to s

$$\frac{du_x}{ds}(x(s), y(s), z(s), t(s)) = \frac{\partial u_x}{\partial x}\frac{dx}{ds} + \frac{\partial u_x}{\partial y}\frac{dy}{ds} + \frac{\partial u_x}{\partial z}\frac{dz}{ds} + \frac{\partial u_x}{\partial t}\frac{dt}{ds} \tag{62}$$

$$\frac{du_y}{ds}(x(s), y(s), z(s), t(s)) = \frac{\partial u_y}{\partial x}\frac{dx}{ds} + \frac{\partial u_y}{\partial y}\frac{dy}{ds} + \frac{\partial u_y}{\partial z}\frac{dz}{ds} + \frac{\partial u_y}{\partial t}\frac{dt}{ds} \tag{63}$$

$$\frac{du_z}{ds}(x(s), y(s), z(s), t(s)) = \frac{\partial u_z}{\partial x}\frac{dx}{ds} + \frac{\partial u_z}{\partial y}\frac{dy}{ds} + \frac{\partial u_z}{\partial z}\frac{dz}{ds} + \frac{\partial u_z}{\partial t}\frac{dt}{ds} \tag{64}$$

$$\tag{65}$$

We take s as

$$\frac{dx}{ds} = u_x \tag{66}$$

$$\frac{dy}{ds} = u_y \tag{67}$$

$$\frac{dz}{ds} = u_z \tag{68}$$

$$\frac{dt}{ds} = 1 \tag{69}$$

Then, eq. (62), (63), (64) becomes

$$\frac{du_x}{ds}(x(s), y, z, t(s)) = \frac{\partial u_x}{\partial x}u_x + \frac{\partial u_x}{\partial y}u_y + \frac{\partial u_x}{\partial z}u_z + \frac{\partial u_x}{\partial t}. \tag{70}$$

$$\frac{du_y}{ds}(x(s), y, z, t(s)) = \frac{\partial u_y}{\partial x}u_x + \frac{\partial u_y}{\partial y}u_y + \frac{\partial u_y}{\partial z}u_z + \frac{\partial u_y}{\partial t}. \tag{71}$$

$$\frac{du_z}{ds}(x(s), y, z, t(s)) = \frac{\partial u_z}{\partial x}u_x + \frac{\partial u_z}{\partial y}u_y + \frac{\partial u_z}{\partial z}u_z + \frac{\partial u_z}{\partial t}. \tag{72}$$

Because of right hand side of this equation equals original advection equation

$$\frac{du_x}{ds}(x(s), y(s), z(s), t(s)) = 0. \tag{73}$$

$$\frac{du_y}{ds}(x(s), y(s), z(s), t(s)) = 0. \tag{74}$$

$$\frac{du_z}{ds}(x(s), y(s), z(s), t(s)) = 0. \tag{75}$$

This means that u_x, u_y, u_z does not depend on s.

Now we can update velocity field from this equation. First, from eq. (69), $t = s + t_0$. We take initial time as $t_0 = 0$, so we have

$$t = s. \tag{76}$$

Second, from eq. (66), (67), (68), we have

$$x(s) = u_x s + x_0 \tag{77}$$
$$y(s) = u_y s + y_0 \tag{78}$$
$$z(s) = u_z s + z_0 \tag{79}$$

where we used the fact that u_x does not depend on s. Third, we integrate eq. (73) from 0 to t

$$\begin{aligned}
u_x(x(t), y(t), z(t), t(t)) &= u_x(x(0), y(0), z(0), t(0)) && (80)\\
&= u_x(x_0, y_0, z_0, 0) && (81)\\
&= \phi_x(x - u_x t, y - u_y t, z - u_z t) && (82)\\
u_y(x(t), y(t), z(t), t(t)) &= \phi_y(x - u_x t, y - u_y t, z - u_z t) && (83)\\
u_z(x(t), y(t), z(t), t(t)) &= \phi_z(x - u_x t, y - u_y t, z - u_z t) && (84)\\
& && (85)
\end{aligned}$$

where we used eq. (76), eq. (77), eq. (78) and eq. (79). In fact, we can confirm that eq. (80) is solution of original advect equation

$$\begin{aligned}
&\frac{\partial \phi(x - u_x t, y - u_y t, z - u_z t)}{\partial t} + u_x \frac{\partial \phi(x - u_x t, y - u_y t, z - u_z t)}{\partial x} && (86)\\
+u_y &\frac{\partial \phi(x - u_x t, y - u_y t, z - u_z t)}{\partial y} + u_z \frac{\partial \phi(x - u_x t, y - u_y t, z - u_z t)}{\partial z} && (87)\\
&= -u_x \frac{\partial \phi}{\partial x} - u_y \frac{\partial \phi}{\partial y} - u_z \frac{\partial \phi}{\partial z} + u_x \frac{\partial \phi}{\partial x} + u_y \frac{\partial \phi}{\partial y} + u_z \frac{\partial \phi}{\partial z} = 0 && (88)
\end{aligned}$$

In the notation of Chapter 2, we have

$$w_{2,x}(x, y, z) = w_{1,x}(x - u_x \Delta t, y - u_y \Delta t, z - u_z \Delta t) \tag{89}$$
$$w_{2,y}(x, y, z) = w_{1,y}(x - u_x \Delta t, y - u_y \Delta t, z - u_z \Delta t) \tag{90}$$
$$w_{2,z}(x, y, z) = w_{1,z}(x - u_x \Delta t, y - u_y \Delta t, z - u_z \Delta t) \tag{91}$$

Appendix B Monotic Cubic Interpolation

In the calculation of advect term, we use interpolation method for estimate velocity in a voxel. A well known interpolation method is Cubic Interpolation. We think a data set consists of data $f_k, k = 0, \ldots, N$. In the method, the value at a point $t \in [t_k, t_k + 1]$ is defined as

$$f(t) = a_3(t - t_k)^3 + a_2(t - t_k)^2 + a_1(t - t_k) + a_0 \tag{92}$$

where

$$a_3 = d_k + d_{k+1} - 2\Delta_k \tag{93}$$
$$a_2 = 3\Delta_k - 2d_k - d_{k+1} \tag{94}$$
$$a_1 = d_k \tag{95}$$
$$a_0 = f_k \tag{96}$$

and

$$d_k = (f_{k+1} - f_{k-1})/2 \tag{97}$$
$$\Delta_k = f_{k+1} - f_k \tag{98}$$

A disadvantage of this interpolation in our simulation is overshoots of data. We modify this method to hold monotonicity in each interval. A necessary condition for monotonicity in each interval is that signs don't change in each interval $sign(f'(t_k)) = sign(f'(t_{k+1})) = sign((f(t_{k+1}) - f(t_k))/(t_{k+1} - t_k))$. This condition is discretized as

$$\begin{cases} sign(d_k) = sign(d_{k+1}) = sign(\Delta_k), & \Delta_k \neq 0 \quad (99) \\ d_k = d_{k+1} = 0, \quad \Delta_k = 0 & . \quad (100) \end{cases}$$

So in our method, we first calculate signs of d_k, d_{k+1}, Δ_k and set these to 0 when their signs are different.

$$d_k = d_{k+1} = \Delta_k = 0 \tag{101}$$

$$when\ sign(d_k) \neq sign(d_{k+1})\ or\ sign(d_k) \neq sign(\Delta_k)\ or\ sign(d_{k+1}) \neq sign(\Delta_k). \tag{102}$$