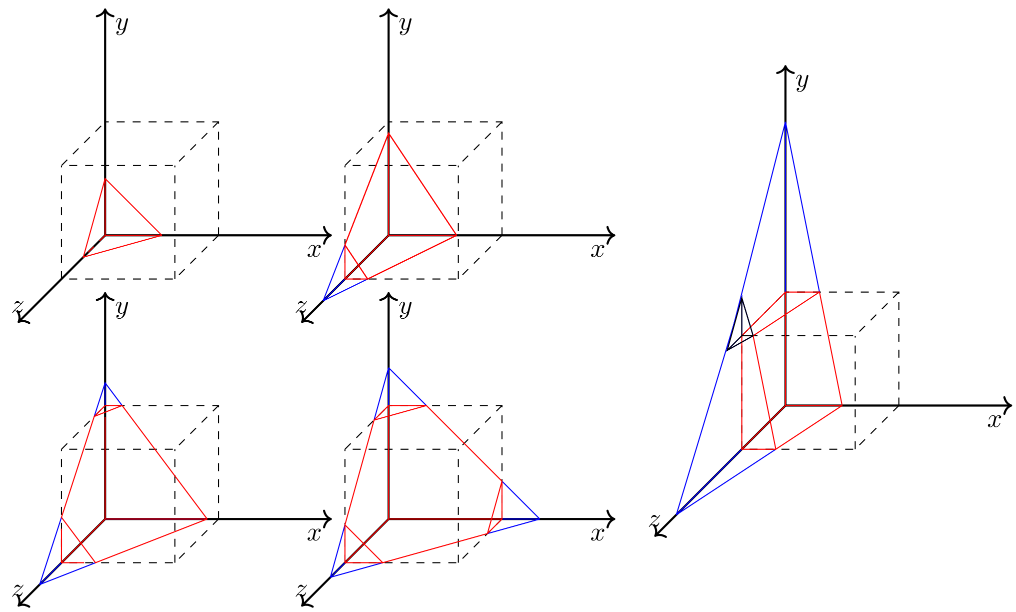

1 Flooding algorithm

For Equations

$$m_1x + m_2y + m_3z = \alpha \tag{1}$$

Supposed it ranks as $m_1 < m_2 < m_3$.

Three intercepts on the coordcates are

$$m_1x + m_2y + m_3z = \alpha \tag{2}$$

$$\begin{aligned} h_1 &= \frac{\alpha}{m_1} \\ h_2 &= \frac{\alpha}{m_2} \\ h_3 &= \frac{\alpha}{m_3} \end{aligned} \tag{3}$$

where $h_1 > h_2 > h_3$.

Based on the

1.1 Cut 1

In this case, $\alpha < m_1$, so that $h_i < 1$, the cutting polyhedron ia a tetrahedron.

$$\begin{aligned} V &= \frac{1}{6}h_1h_2h_3 \\ &= \frac{\alpha}{6m_1m_2m_3} \end{aligned} \tag{4}$$

$$c_i = \frac{1}{4}h_i = \frac{\alpha}{m_i} \tag{5}$$

1.2 Cut 2

In this case, $m_1 \leq \alpha < m_2$, $h_1 > 1$, $h_2 < 1$ and $h_3 < 1$. The cutting plane is a quadrilateral and the polyhedron can be expressed as a larger tetrahedron minus a smaller tetrahedron

$$\begin{aligned} V &= \frac{1}{6}h_1h_2h_3\left(1 - \frac{(h_1-1)^3}{h_1^3}\right) \\ &= \frac{1}{6}h_1h_2h_3\left(\frac{3h_1^2 - 3h_1 + 1}{h_1^3}\right) \\ &= \frac{h_2h_3}{6h_2^2} + \frac{(h_1-1)h_2h_3}{h_1} \\ &= \frac{\alpha}{6m_1m_2m_3} \end{aligned} \tag{6}$$

$$\begin{aligned} V &= \frac{1}{6}h_1h_2h_3\left(1 - \frac{(h_1-1)^3}{h_1^3}\right) \\ &= \frac{1}{6}h_1h_2h_3\left(\frac{3h_1^2 - 3h_1 + 1}{h_1^3}\right) \\ &= \frac{h_2h_3}{6h_1^2} + \frac{(h_1-1)h_2h_3}{h_1} \\ &= \frac{m_1}{6m_2m_3} + \frac{\alpha(\alpha - m_1)}{m_2m_3} \end{aligned} \tag{7}$$

$$\begin{aligned} V &= \frac{1}{6}h_1h_2h_3\left(1 - \frac{(h_1-1)^3}{h_1^3}\right) \\ &= \frac{1}{6}h_1h_2h_3\left(\frac{3h_1^2 - 3h_1 + 1}{h_1^3}\right) \\ &= \frac{h_2h_3}{6h_1^2} + \frac{(h_1-1)h_2h_3}{h_1} \\ &= \frac{m_1}{6m_2m_3} + \frac{\alpha(\alpha - m_1)}{m_2m_3} \end{aligned} \tag{8}$$

2 Advection of Volume function

Use a simple 3 stencil 1D grid as an example:

2.1 Weymouth-Yue

Original volume plus the boundary flux (Eulerian).

$$\tilde{f}_c = f_c + VOF_c^1 - VOF_c^3 - VOF_r^1 + VOF_l^3 \tag{9}$$

2.2 CIAM

Backward lagrangian of the grid face and find the intersection between two faces (Lagrangian).

-

$$\tilde{f}_c = VOF_c^2 + VOF_r^1 + VOF_l^3 \tag{10}$$

Compared with W-Y advection, we obtain

$$VOF2_c = f_c - 2VOF_r^1 - VOF_c^1 - VOF_c^3 \tag{11}$$

3 Find the best centroid in MOF

4 Optimization algorithm

For a minimization problem

$$x = \arg\min_x \frac{1}{2}\|f(x)\|^2 \tag{12}$$

expand with Taylor series and obtains

$$\frac{1}{2}\|f(x+\Delta x)\|^2 = \frac{1}{2}\left\{f(x)^T f(x) + 2f(x)^T J(x)\Delta x + \Delta x^T J(x)^T J(x)\Delta x\right\} \tag{13}$$

Where $J(x)$ is the Jacobian matrix. Take the derivative on both side and obtains

$$\nabla f(X) = \nabla f\left(X_{i+1}\right) + H_{i+1}\left(X - X_{i+1}\right) \tag{14}$$

Where is the Hessian matrix and it will be easy to calculate the $\nabla f(X_i) = J^T f(X_i)$.

4.1 Gauss-Newton algorithm

Uses an approximate Hessian matrix

$$H = J^T J \tag{15}$$

Now it would be easy to calculate the step $\Delta x = X_{i+1} = X_i$ by

$$\Delta x_i = -H_{i+1}^T \nabla f(X_{i-1}) = -(J^T J)(J^T X) \tag{16}$$

This equation can be use for Gauss-Newton iteration.

4.2 Davidon-Fletcher-Powel (DFP) algorithm

The DFP algorithm do not calculate the inverse of the Hessian matrix, instead, is uses an approximate inverse Hessian.

Assume the Hessian matrix can be calculated from the Hessian matrix of the previous step,

$$H_{i+1} = H_i + E_i \tag{17}$$

Let

$$E_i = mvv^T + nww^T$$

(18)

When m, n, v are determined, H_i is obtained.

let $s_i = X_{i+1} - X_i$, $y_i = \nabla X_{i+1} - \nabla X_i$, A direct form of the DFP algorithm is

$$H_{k+1}^{-1} = H_k^{-1} - \frac{H_k^{-1} y_k y_k^T H_k^{-1}}{y_k^T H_k^{-1} y_k} + \frac{s_k s_k^T}{y_k^T s_k} \tag{19}$$

4.3 Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm

An additional term is included in BFGS compared with DFP. The expression for new Hessian inverse is

$$H_{k+1}^{-1} = H_k^{-1} + \frac{\left(\mathbf{s}_k^{\mathrm{T}} \mathbf{y}_k + \mathbf{y}_k^{\mathrm{T}} H_k^{-1} \mathbf{y}_k\right)\left(\mathbf{s}_k \mathbf{s}_k^{\mathrm{T}}\right)}{\left(\mathbf{s}_k^{\mathrm{T}} \mathbf{y}_k\right)^2} - \frac{H_k^{-1} \mathbf{y}_k \mathbf{s}_k^{\mathrm{T}} + \mathbf{s}_k \mathbf{y}_k^{\mathrm{T}} H_k^{-1}}{\mathbf{s}_k^{\mathrm{T}} \mathbf{y}_k} \tag{20}$$

4.4 LevenBerg-Marquardt (LM) algorithm