以下のように定める．（定義不足だが，適宜うまく読んで下さい．）

$$
\begin{aligned}
\vec{u} &= [u, v]^T = [u(x,y), v(x,y)]^T \\
\vec{c} &= [a, b]^T = \text{const.}\ (\text{vector field})
\end{aligned}
$$

ベクトル場 [a, b]^T に沿う [u, v]^T の移流方程式は，

$$
\begin{aligned}
\frac{\partial \vec{u}}{\partial t} + \left(\vec{c} \cdot \left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right]^T \right) \vec{u} = 0 \\
\frac{\partial \vec{u}}{\partial t} + \left(a\frac{\partial}{\partial x} + b\frac{\partial}{\partial y} \right) \vec{u} = 0 \\
\frac{\partial \vec{u}}{\partial t} + a\frac{\partial \vec{u}}{\partial x} + b\frac{\partial \vec{u}}{\partial y} = 0 \\
\left[\frac{\partial u}{\partial t}, \frac{\partial v}{\partial t} \right]^T + a\left[\frac{\partial u}{\partial x}, \frac{\partial v}{\partial x} \right]^T + b\left[\frac{\partial u}{\partial y}, \frac{\partial v}{\partial y} \right]^T = 0
\end{aligned}
$$

なので，成分ごとに書くと，

$$
\begin{aligned}
\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} = 0 \\
\frac{\partial v}{\partial t} + a\frac{\partial v}{\partial x} + b\frac{\partial v}{\partial y} = 0
\end{aligned}
$$

これを差分化（1次の風上差分法）すると，（vについては省略．）

$$
\begin{aligned}
\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} + a\frac{u_{i+1,j}^n - u_{i-1,j}^n}{2\Delta x} + |a|\frac{-u_{i+1,j}^n + 2u_{i,j}^n - u_{i-1,j}^n}{2\Delta x} \\
+ b\frac{u_{i,j+1}^n - u_{i,j-1}^n}{2\Delta y} + |b|\frac{-u_{i,j+1}^n + 2u_{i,j}^n - u_{i,j-1}^n}{2\Delta y} = 0
\end{aligned}
$$

となる．（非粘性）バーガース方程式については，

$$
\vec{c} := \vec{u}
$$

とすれば良いので，（2次元）バーガース方程式についての差分法は，

$$
\begin{aligned}
\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta t} + u_{i,j}^n\frac{u_{i+1,j}^n - u_{i-1,j}^n}{2\Delta x} + |u_{i,j}^n|\frac{-u_{i+1,j}^n + 2u_{i,j}^n - u_{i-1,j}^n}{2\Delta x} \\
+ v_{i,j}^n\frac{u_{i,j+1}^n - u_{i,j-1}^n}{2\Delta y} + |v_{i,j}^n|\frac{-u_{i,j+1}^n + 2u_{i,j}^n - u_{i,j-1}^n}{2\Delta y} = 0
\end{aligned}
$$

となる．（vの時間発展も同様の差分を取れば良い．）

コードに落とし込むと，移流の方は，c = [ad1, ad2]^T = [1, 1]と見れば，

```
void upWind_firstOrder_X(Array& u, Array& v, Array& u_next){
    double ad1, ad2;
    for(int i=0; i<Nx; ++i){
        for(int j=0; j<Ny; ++j){
            ad1 = 1.0;
            ad2 = 1.0;
            u_next[i][j] = u[i][j]
                             - dt * ( ad1 * (u[i+1][j] - u[i-1][j]) / (2.0 * dx)
+ std::abs(ad1) * (- u[i+1][j] + 2.0 * u[i][j] - u[i-1][j]) / (2.0 * dx))
```

```
                                - dt * ( ad2 * (u[i][j+1] - u[i][j-1]) / (2.0 * dy)
+ std::abs(ad2) * (- u[i][j+1] + 2.0 * u[i][j] - u[i][j-1]) / (2.0 * dy));
        }
    }
}

void upWind_firstOrder_Y(Array& u, Array& v, Array& v_next){
    double ad1, ad2;
    for(int i=0; i<Nx; ++i){
        for(int j=0; j<Ny; ++j){
            ad1 = 1.0;
            ad2 = 1.0;
            v_next[i][j] = v[i][j]
                                - dt * ( ad1 * (v[i+1][j] - v[i-1][j]) / (2.0 * dx)
+ std::abs(ad1) * (- v[i+1][j] + 2.0 * v[i][j] - v[i-1][j]) / (2.0 * dx))
                                - dt * ( ad2 * (v[i][j+1] - v[i][j-1]) / (2.0 * dy)
+ std::abs(ad2) * (- v[i][j+1] + 2.0 * v[i][j] - v[i][j-1]) / (2.0 * dy));
        }
    }
}
```

バーガース方程式の方は,

```
void upWind_firstOrder_X(Array& u, Array& v, Array& u_next){
    double ad1, ad2;
    for(int i=0; i<Nx; ++i){
        for(int j=0; j<Ny; ++j){
            ad1 = u[i][j];
            ad2 = v[i][j];
            u_next[i][j] = u[i][j]
                                - dt * ( ad1 * (u[i+1][j] - u[i-1][j]) / (2.0 * dx)
+ std::abs(ad1) * (- u[i+1][j] + 2.0 * u[i][j] - u[i-1][j]) / (2.0 * dx))
                                - dt * ( ad2 * (u[i][j+1] - u[i][j-1]) / (2.0 * dy)
+ std::abs(ad2) * (- u[i][j+1] + 2.0 * u[i][j] - u[i][j-1]) / (2.0 * dy));
        }
    }
}

void upWind_firstOrder_Y(Array& u, Array& v, Array& v_next){
    double ad1, ad2;
    for(int i=0; i<Nx; ++i){
        for(int j=0; j<Ny; ++j){
            ad1 = u[i][j];
            ad2 = v[i][j];
            v_next[i][j] = v[i][j]
                                - dt * ( ad1 * (v[i+1][j] - v[i-1][j]) / (2.0 * dx)
+ std::abs(ad1) * (- v[i+1][j] + 2.0 * v[i][j] - v[i-1][j]) / (2.0 * dx))
                                - dt * ( ad2 * (v[i][j+1] - v[i][j-1]) / (2.0 * dy)
+ std::abs(ad2) * (- v[i][j+1] + 2.0 * v[i][j] - v[i][j-1]) / (2.0 * dy));
        }
    }
}
```

となる.

mino2357