Lagrangian particle tracking experiment tutorial

Jang-Geun Choi

Center for Ocean Engineering, University of New Hampshire, NH, USA

Table of Contents

1. Governing equation

Particle tracking experiment is one of the simplest transport model but powerful tool that can be applicable to many problems. The governing equations is given by

$$\frac{d\vec{X}}{dt} = \vec{u}$$

In case of two-dimensional problem, $\vec{X} = (X, Y)$ and $\vec{u} = (u, v)$. Note that equation above is a system of equations that consists of two ordinary differential equations ($dX/dt = u$ and $dY/dt = v$). It is not hard to solve the system of equation using numerical approach. Based on first order forward Euler scheme, the equations can be discretized to

$$\frac{X^{i+1} - X^i}{\Delta t} = u^i$$

$$\frac{Y^{i+1} - Y^i}{\Delta t} = v^i$$

so

$$X^{i+1} = X^i + \Delta t\, u^i$$

$$Y^{i+1} = Y^i + \Delta t\, v^i$$

where upper script i and $i+1$ represents variables in current and nex step in time, respectively. Therefore, based on given initial position of position (X^0 and Y^0) and given velocity fields (u and v), position in first step ($t = \Delta t$) can be calculated, and then that of second step ($t = 2\Delta t$) can be calculated using the previous step. This can be continued.

2. Practice for model development: numerical error and scheme

Consider very simple velocity fields given by

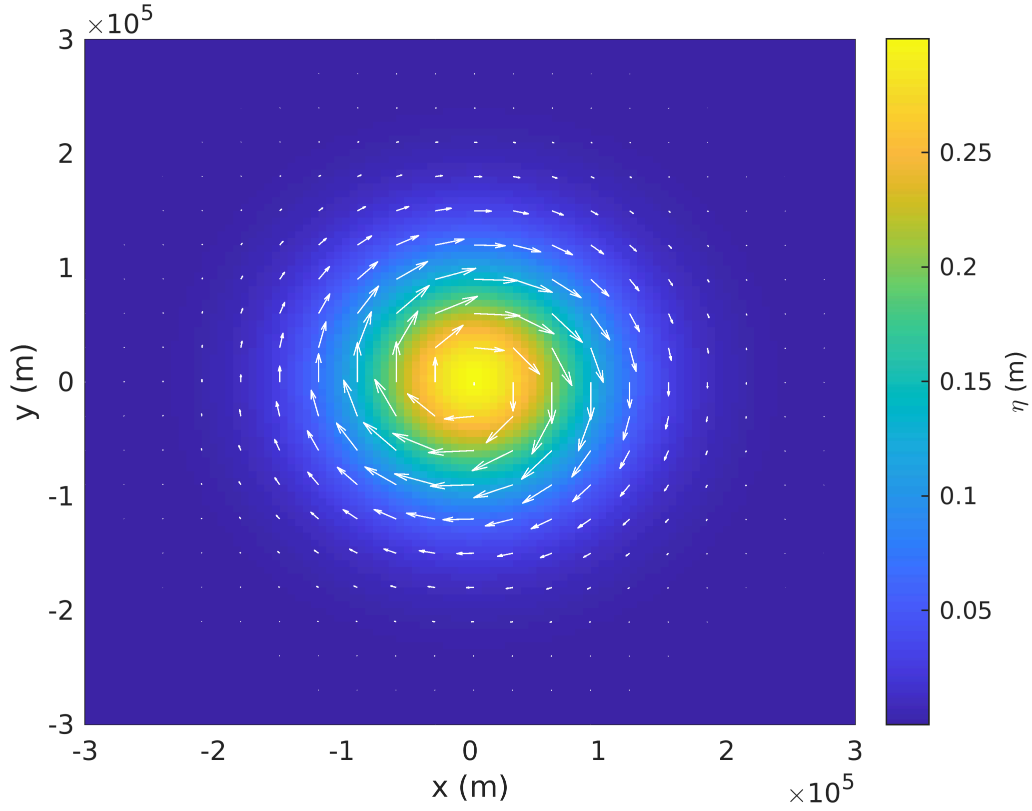
$$\eta = \eta_0 e^{-(x^2+y^2)/L^2}$$

$$v = \frac{g}{f}\frac{\partial \eta}{\partial x} = -\frac{g\eta_0}{f}\frac{2x}{L^2}e^{-(x^2+y^2)/L^2}$$

$$u = -\frac{g}{f}\frac{\partial\eta}{\partial y} = \frac{g\eta_0}{f}\frac{2y}{L^2}e^{-(x^2+y^2)/L^2}$$

where $\eta_0 = 0.3\,m$, $g = 10\,m\,s^{-2}$, $f = 10^{-4}\,s^{-1}$, $L = 10^5\,m$. This analytical equations describes simple clockwise eddy governed by pure geostrophic current.

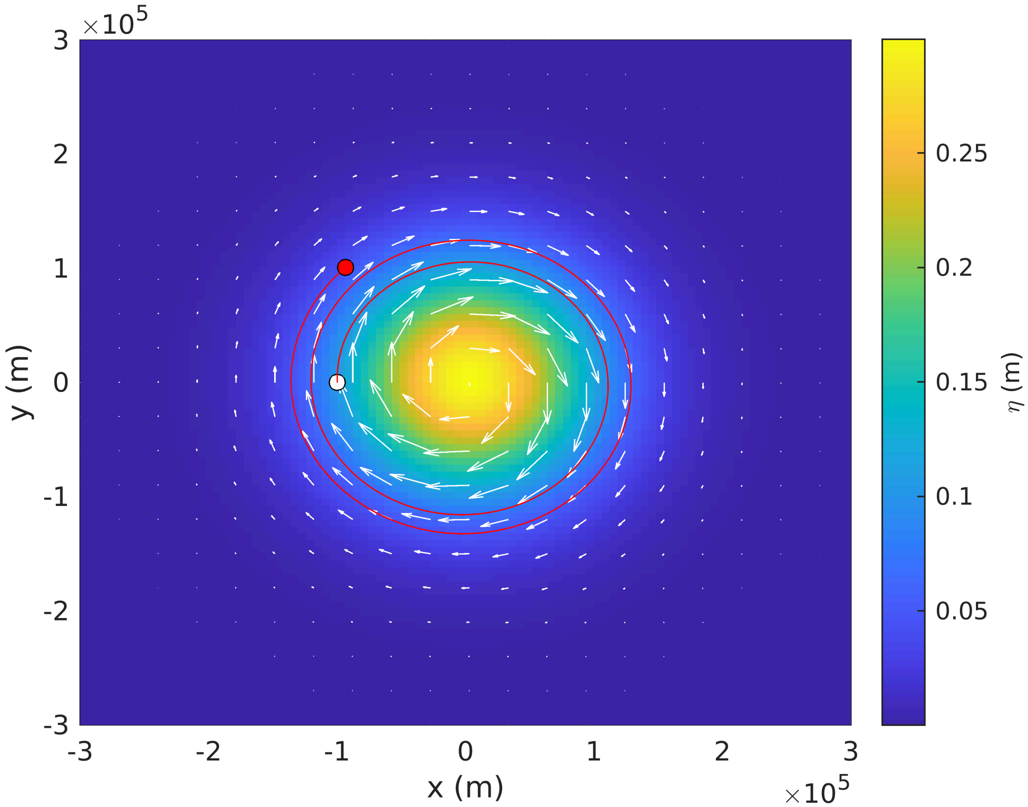
```
clc;clear;close all;

eta0=0.3;
g=10;
f=1e-4;
L=1e5;

x1=linspace(-3*L,3*L,100);
y1=linspace(-3*L,3*L,101);
[y,x]=meshgrid(y1,x1);

eta=eta0*exp(-(x.^2+y.^2)/L^2);
v=-g/f*2*x/L^2.*exp(-(x.^2+y.^2)/L^2);
u=g/f*2*y/L^2.*exp(-(x.^2+y.^2)/L^2);

pcolor(x,y,eta)
hold on
nn=5;
quiver(x(1:nn:end,1:nn:end),y(1:nn:end,1:nn:end),...
    u(1:nn:end,1:nn:end),v(1:nn:end,1:nn:end),'w')
shading flat
xlabel('x (m)')
ylabel('y (m)')
cb=colorbar;
ylabel(cb,'\eta (m)')
```



As initial condition (location) of particle, consider $X^0 = -1\times 10^5\, m$ and $Y^0 = 0\, m$. Position at the next step can be calculated by $X^{i+1} = X^i + \Delta t\, u$ and $Y^{i+1} = Y^i + \Delta t\, v$ mentioned above where time step will be sett to $\Delta t = 10^3\, s$ and 100 steps will be calculated. It is worth noting that u and v can be interpolated from given fields.

```
dt=1e4;
X0=-1e5;
Y0=0;

n=3000000/dt;
T=(0:dt:n*dt)';
X=NaN(size(T));
Y=NaN(size(T));

X(1)=X0;
Y(1)=Y0;

for i=1:n
    ui=interp2(y,x,u,Y(i),X(i));
    vi=interp2(y,x,v,Y(i),X(i));

    X(i+1)=X(i)+ui*dt;
    Y(i+1)=Y(i)+vi*dt;
end

hold on
plot(X0,Y0,'ok','MarkerFaceColor','w')
```

```
plot(X,Y,'r')
plot(X(end),Y(end),'ok','MarkerFaceColor','r')
```



Note that the particle slowly move down and η experienced by the particle decrease as time goes. This is numerical error. To be specific, the equations for velocity field satisfying

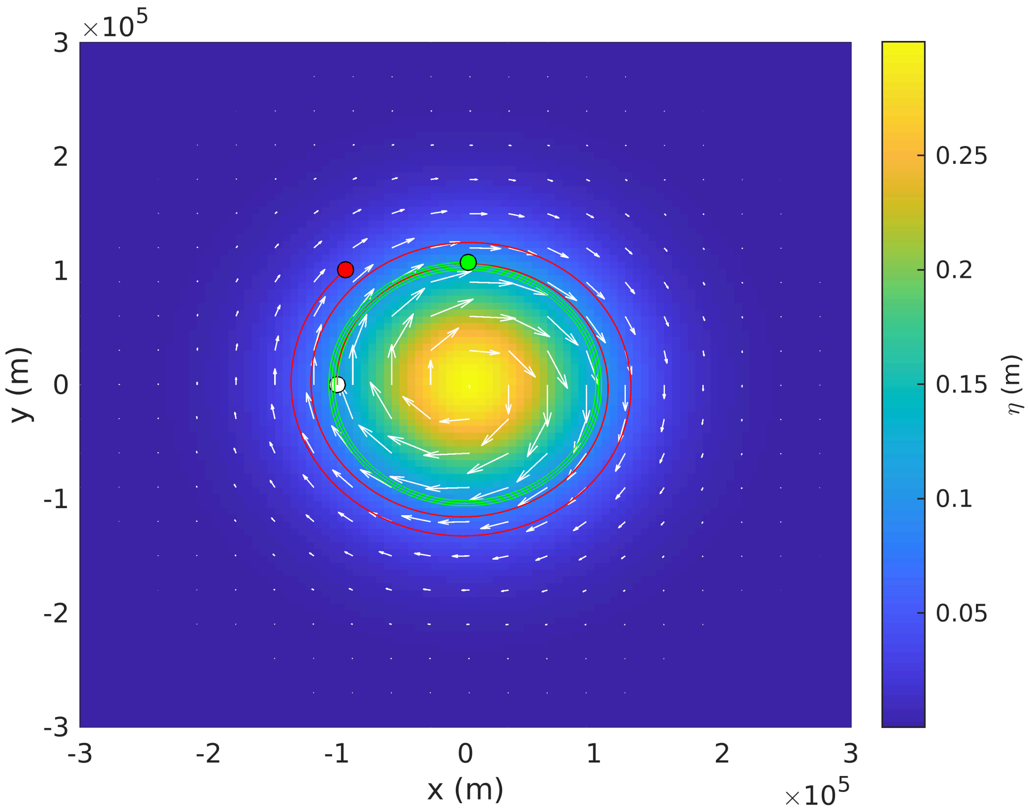
$$\frac{\partial \eta}{\partial t} + u\frac{\partial \eta}{\partial x} + v\frac{\partial \eta}{\partial y} = 0 \leftrightarrow \frac{D\eta}{Dt} = 0.$$

It can be mathematically verified by substituting the equations (definitions) for η, u, and v into the equation above. The equation above means that value of η should not be changed as time in terms of Lagrangian point of view. Therefore, analytically, particle follows contour-line of η to conserve the value.

The numerical error can be reduced by choosing smaller time step.

```
clear X Y

dt=1e3;

n=3000000/dt;
T=(0:dt:n*dt)';
X=NaN(size(T));
Y=NaN(size(T));

X(1)=X0;
Y(1)=Y0;
```

```
for i=1:n
    ui=interp2(y,x,u,Y(i),X(i));
    vi=interp2(y,x,v,Y(i),X(i));

    X(i+1)=X(i)+ui*dt;
    Y(i+1)=Y(i)+vi*dt;
end

plot(X,Y,'g')
plot(X(end),Y(end),'ok','MarkerFaceColor','g')
```



Another way to reduce the error is to use better (higher order) numerical scheme. Based on Heun's scheme, the governing equation for particle position is discretized to

$$\widetilde{X} = X^i + \Delta t u(X^i, Y^i)$$

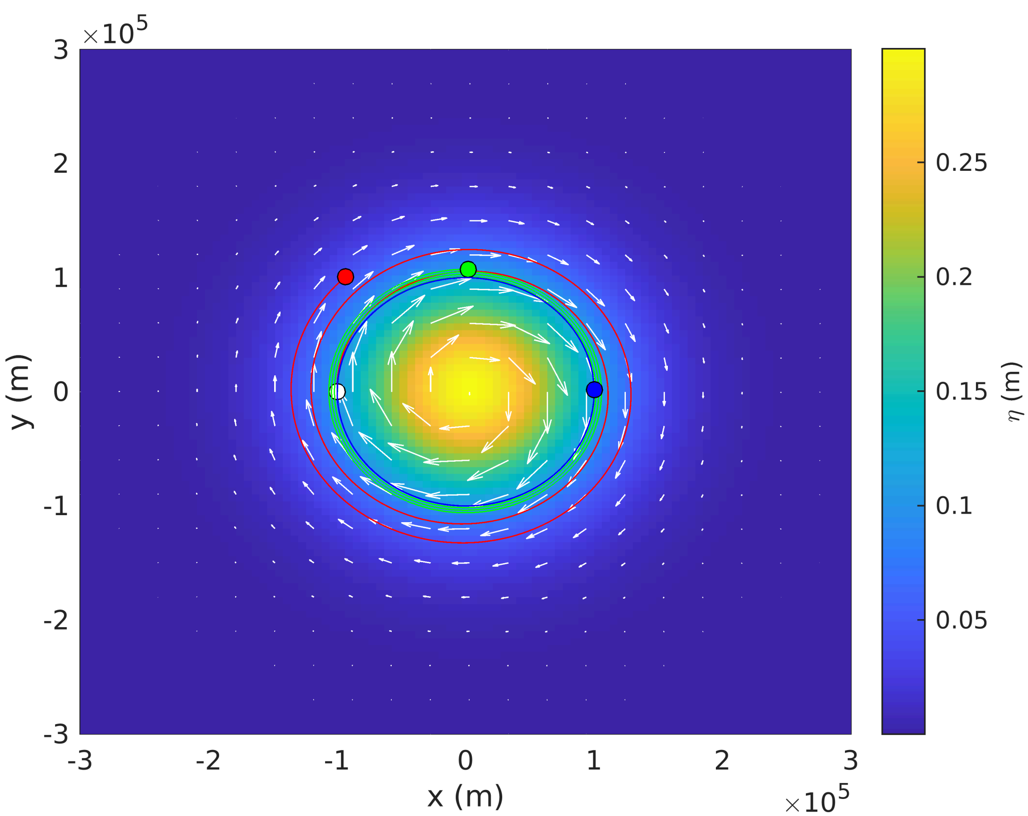
$$\widetilde{Y} = Y^i + \Delta t u(X^i, Y^i)$$

$$X^{i+1} = X^i + \Delta t(u(X^i, Y^i) + u(\widetilde{X}, \widetilde{Y}))/2$$

$$Y^{i+1} = Y^i + \Delta t(v(X^i, Y^i) + v(\widetilde{X}, \widetilde{Y}))/2$$

```
clear X Y
```

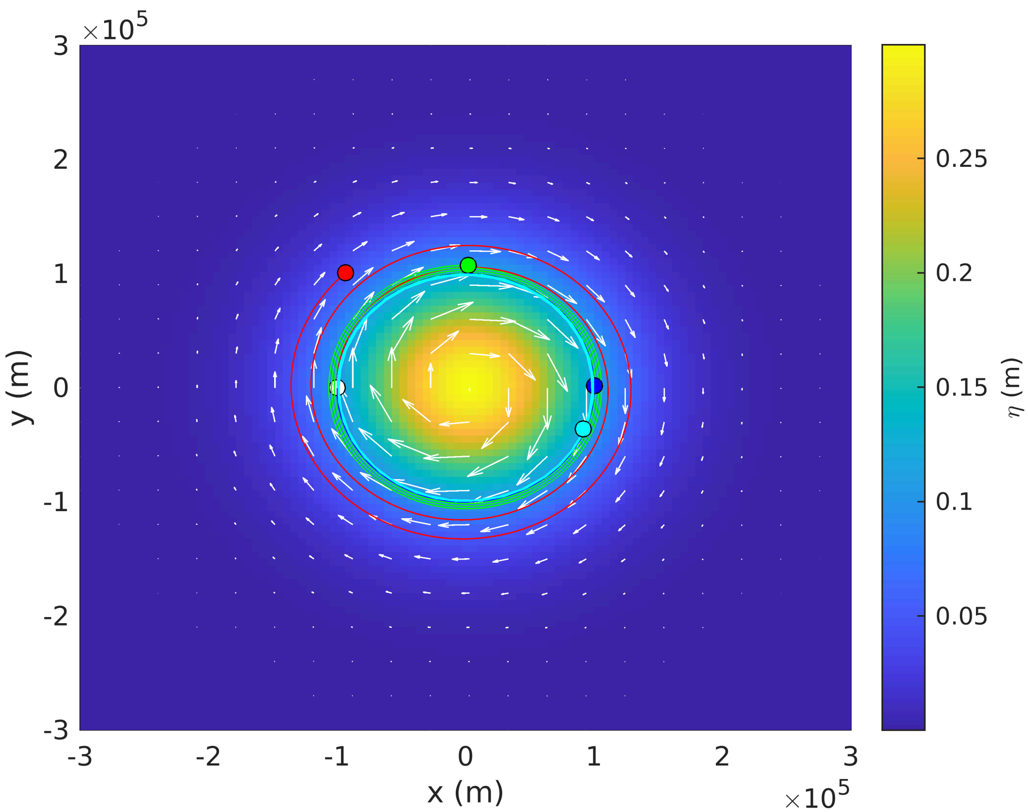
```
dt=1e4;

n=3000000/dt;
T=(0:dt:n*dt)';
X=NaN(size(T));
Y=NaN(size(T));

X(1)=X0;
Y(1)=Y0;

for i=1:n
    ui1=interp2(y,x,u,Y(i),X(i));
    vi1=interp2(y,x,v,Y(i),X(i));

    Xi=X(i)+ui1*dt;
    Yi=Y(i)+vi1*dt;

    ui2=interp2(y,x,u,Yi,Xi);
    vi2=interp2(y,x,v,Yi,Xi);

    X(i+1)=X(i)+(ui1+ui2)/2*dt;
    Y(i+1)=Y(i)+(vi1+vi2)/2*dt;
end

plot(X,Y,'b')
plot(X(end),Y(end),'ok','MarkerFaceColor','b')
```



Note that this scheme has better accuracy (much more conservative η) regardless of 10 times longer time step. Note that n-th order scheme indicates error is proportional to $(\Delta t)^n$. Higher order scheme indicates more rapid decrease of error as Δt decrease.

MATLAB provides various ordinary equation solvers (https://www.mathworks.com/help/matlab/ordinary-differential-equations.html). Runge-Kutta scheme is one of the most famous solver and frequently adopted to particle tracking experiments. Below is an example code for application of the scheme using MATLAB built-in function. It is worth noting that a guidance for choosing solver is provided by MathWorks (https://www.mathworks.com/help/matlab/math/choose-an-ode-solver.html).

```
clear X Y

X(1)=X0;
Y(1)=Y0;

[t,XY]=ode45(@(t,Xv)uv(t,Xv,x,y,u,v),[0 3000000],[X Y]);
X=XY(:,1);
Y=XY(:,2);

plot(X,Y,'c')
plot(X(end),Y(end),'ok','MarkerFaceColor','c')
```




```
function dXdt = uv(~,X,x,y,u,v)
```

```
dXdt(1) = interp2(y,x,u,X(2),X(1));
dXdt(2) = interp2(y,x,v,X(2),X(1));
dXdt=dXdt';
end
```