

This file contains the mathematical derivation  
and explanations of corresponding code at  
<https://github.com/xinan-nancy-chen/rOMT>.

Created by Xinan Chen at 04/13/21 and  
finished at 04/16/21.

I apologize for my poor handwriting if it causes  
your trouble to read.

A LaTeX version is expected in the future.

rOMT code. written by Xian Chen on 04/13/21.

romt: regularized optimal mass transport.

Question in continuous version:

$$\begin{cases} \min_{\rho, v} \int_0^1 \int_{\Omega} \rho(t, x) \|v(t, x)\|^2 dx dt \\ \text{s.t. } \begin{cases} \rho_t + \nabla \cdot (\rho v) = \delta \Delta \rho & \textcircled{1} \\ \rho(0, x) = \rho_0(x), \quad \rho(1, x) = \rho_1(x) \end{cases} \end{cases}$$

where  $\rho(t, x)$ : density function at location  $x$  and time  $t$

$v(t, x)$ : velocity field

① advection-diffusion equation.

Question in discrete version:

$$\begin{cases} \min_{v} J(v) = \beta \cdot \text{hd} \cdot dt \cdot \rho^T \left( I_m \otimes [I_n | I_n | I_n] \right) (v \odot v) \\ \text{s.t. } \begin{cases} \frac{1}{2} \|\rho_m - \rho_{\text{obs}}\|^2 + \frac{1}{2} \gamma \cdot \text{hd} \cdot dt \|\text{Grad } v\|^2 \\ \text{matching final observed image} \\ (I_n - dt \cdot Q) \rho_{i+1} = S(v_i) \rho_i, \quad i=0, \dots, m-1. \text{ (*)} \\ \rho_0 = \rho_{\text{obs}} \end{cases} \end{cases}$$

denote dimension of matrix/vector  
where  $\rho = \begin{pmatrix} \rho_1 \\ \vdots \\ \rho_m \end{pmatrix}$  vertical vector

$v = \begin{pmatrix} v_0 \\ \vdots \\ v_{m-1} \end{pmatrix}$  vertical vector

⊗: Kronecker ; ∘: Hadamard product.

The discretization is

for space, it is a  $N_x \times N_y \times N_z$  cube of same length.  $\text{hd}$  = volume of each sub-cube  
(in our code, we use  $\text{hd} = 1 \times 1 \times 1 = 1$ ).

in total  $n = N_x \cdot N_y \cdot N_z$  cells,

for time, it is divided into  $m$  equal intervals of length  $dt$ .  
(in this note, we use "m" to replace "nt" sometimes).

$t=0$	$t=dt$	$t=2dt$	...	$t=(m-1)dt$	$t=m dt$
$\rho_0 = \rho_0^{\text{obs}}$ (given)	$v_0$	$v_1$	$v_2$	$v_{m-2}$	$v_{m-1}$

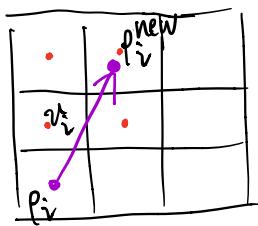
From  $\rho_0 = \rho_0^{\text{obs}}$  and  $v$ , we can derive  $\rho$  by (\*) (corresponds to advecDiff.m in code).

in (\*), it's the discretization of equation ①, which is a process of advection + diffusion.

So we solve it numerically as first advection, second diffusion.

for  $i$ , advection:  $\rho_t + \nabla \cdot (\rho v) = 0 \Rightarrow \rho_i^{\text{adv}} = S(v_i) \rho_i$ , where  $S(v_i)$  is the movement of  $\rho$  according to  $v$ . averaging matrix w.r.t  $v_i$

①



(corresponds to aTrilinear.m in code)

Particle in cell method:

move  $p_i$  according to  $v_i$  to get new location  $p_i^{\text{new}}$   
then split  $p_i^{\text{new}}$  into its nearest neighbor grid  
points by a ratio determined by distance,  
i.e., the farther distance, the smaller the ratio.

After that, we get  $p_i^{\text{adv}}$  which is the distribution  
after advection.

$$\text{diffusion: } \rho_t = 6\Delta p \Rightarrow \frac{p_{ith} - p_i^{\text{adv}}}{dt} = Q p_{ith} \xrightarrow{\text{Backwards scheme}}$$

$Q$  is discretization of  $6\Delta$  on grid.

$$\Rightarrow (I_n - dt Q) p_{ith} = p_i^{\text{adv}}$$

So combine the two steps together, we have  $(*)$ .

Denote  $\overset{n}{L} = I_n - dt Q$ .

From  $(*)$  we have  $p_{ith} = \overset{k}{L} S(v_i) p_i$ ,  $i=0, \dots, m-1$ .

So for  $k=1, \dots, m$ .

$$p_k = \overset{k}{L} S(v_{k-1}) \overset{k}{L} S(v_{k-2}) \cdots \overset{k}{L} S(v_0) p_0$$

$$\Rightarrow p_k = p_k(v_0, \dots, v_{k-1}). \rightarrow \text{w.r.t } v_0, \dots, v_{k-1}.$$

②

For energy functional  $P(u)$ , it has three terms,

$$P(u) = \beta P_1(u) + P_2(u) + \gamma P_3(u).$$

$$P_1(u) \sim \rho \|u\|^2.$$

$$\rho = \begin{pmatrix} \rho_1 \\ \vdots \\ \rho_m \end{pmatrix} = \begin{pmatrix} \rho_1 \\ \vdots \\ \rho_m \\ mn \end{pmatrix} \quad \Rightarrow P_1(u) = \text{hd. oft. } \sum_{i=1}^m \rho_i \left( u_{i,x}^2 + u_{i,y}^2 + u_{i,z}^2 \right)$$

$$u = \begin{pmatrix} u_0 \\ \vdots \\ u_{mn} \end{pmatrix} = \begin{pmatrix} u_0 \\ \vdots \\ u_{mn} \\ 3mn \end{pmatrix} = \begin{pmatrix} u_0 \\ \vdots \\ u_{mn} \\ 3mn \end{pmatrix} = \begin{pmatrix} u_0 \\ \vdots \\ u_{mn} \\ 3mn \end{pmatrix} = \begin{pmatrix} u_0 \\ \vdots \\ u_{mn} \\ 3mn \end{pmatrix}$$

(3)

We solve this optimization problem with Newton-Gaussian method:

$$g(w) = \nabla P(w) = \frac{\partial P(w)}{\partial w} \rightarrow \text{gradient of } P(w)$$

$$H(w) = \begin{pmatrix} \frac{\partial^2 P(w)}{\partial w_i \partial w_j} \end{pmatrix}_{i,j} \rightarrow \text{Hessian matrix of } P(w)$$

### Algorithm

```

w = initial guess;
for i=1,..,MaxIter
    Compute g(w), H(w);
    Solve Hs = -g for s;
    Do line search to find length l;
    w = w + ls;
end

```

Next we look into how to compute  $g$  and  $H$ .

recall that

$$\begin{aligned}
P(w) &= \beta \cdot \text{hd} \cdot \text{dt} \underbrace{P^T \left( I_m \otimes [I_n | I_n | I_n] \right)}_{mn} \underbrace{(w \odot w)}_{3mn} \\
&\quad + \frac{1}{2} \| P_m - P_1^{\text{obs}} \|^2 + \frac{1}{2} \gamma \cdot \text{hd} \cdot \text{dt} \| \text{Grad } w \|^2 \\
&= \beta P_1(w) + P_2(w) + \gamma P_3(w)
\end{aligned}$$

(4)

$$\Rightarrow g(v) = \frac{\partial P(v)}{\partial v} = \beta \frac{\partial P_1(v)}{\partial v} + \frac{\partial P_2(v)}{\partial v} + \gamma \frac{\partial P_3(v)}{\partial v}.$$

1°.  $\frac{\partial P_i(v)}{\partial v} = \text{hd}\cdot\text{dt} \frac{\partial}{\partial v} (P^T M(v \otimes v))$  where we let  $M = I_m \otimes [I_n | I_n | I_n]$

$\begin{bmatrix} v \\ v \\ \vdots \\ v \end{bmatrix}$   $\begin{bmatrix} 3mn & 1 \\ 1 & mn \\ mn & 3mn \\ \vdots & \vdots \\ mn & 3mn \end{bmatrix}$

 $= \text{hd}\cdot\text{dt} P^T \nabla [M(v \otimes v)] + (M(v \otimes v))^T \nabla P$ 
 $= \text{hd}\cdot\text{dt} \left[ 2 P^T M \text{diag}(v) + (M(v \otimes v))^T J \right]$ 

where  $J = \begin{pmatrix} J_{v_0}^1 & & & \\ J_{v_0}^2 & J_{v_1}^2 & & \\ \vdots & \vdots & \ddots & \\ J_{v_0}^m & J_{v_1}^m & \cdots & J_{v_{m-1}}^m \end{pmatrix}$ ,  $J_{v_j}^k = \frac{\partial p_k}{\partial v_j}$ ,  $k=1, \dots, m$ ,  $j=0, \dots, m-1$ .

Since  $p_k = p_k(v_0, \dots, v_{k-1})$ ,  $\Rightarrow J_{v_j}^k = 0$  if  $j \geq k$

$\Rightarrow J$  is lower triangular.

$$\text{For } j \leq k-1, J_{v_j}^k = \frac{\partial}{\partial v_j} \left( L^{-1} S(v_{k-1}) L^{-1} S(v_{k-2}) \cdots L^{-1} S(v_0) p_0 \right)$$
 $= \underbrace{L^{-1} S(v_{k-1}) \cdots L^{-1} S(v_{j+1})}_{n} \underbrace{L^{-1} \frac{\partial}{\partial v_j} (S(v_j) L^{-1} S(v_{j+1}) \cdots L^{-1} S(v_0) p_0)}_{3n}.$ 
 $= \underbrace{L^{-1} S(v_{k-1}) \cdots L^{-1} S(v_{j+1})}_{n} \underbrace{L^{-1} \frac{\partial}{\partial v_j} (S(v_j) p_j)}_{3n}$

Since  $S(v_j)$  is linear to  $v_j$ , then  $\frac{\partial}{\partial v_j} (S(v_j) p_j)$  only depends on  $p_j$ .

see details later in dTrilinear.m.



So we denote  $\frac{\partial}{\partial v_j} (S(v_j) p_j) \triangleq B(p_j)$ .

$$\Rightarrow J_{v_j}^k = L^\top S(v_{k+1}) \cdots L^\top S(v_{j+1}) L^\top B(p_j).$$

Thus we have  $J_{v_j}^{k+1} = L^\top S(v_k) J_{v_j}^k$ .

Thus the second term of  $\frac{\partial P_i(u)}{\partial u}$ , which is  $(M(u \circ u))_m^T J_{3mn}$

$$\begin{aligned}
 &= \left[ M_{mn} \underbrace{\begin{pmatrix} v_{0,x}^2 \\ v_{0,y}^2 \\ v_{0,z}^2 \\ \vdots \\ v_{m-1,x}^2 \\ v_{m-1,y}^2 \\ v_{m-1,z}^2 \end{pmatrix}}_{3mn} \right]^T J \\
 &= \left[ \begin{matrix} I_n & I_n & I_n \\ & I_n & I_n & I_n \\ & & \ddots & & \end{matrix} \right] \left[ \begin{matrix} v_{0,x}^2 \\ v_{0,y}^2 \\ v_{0,z}^2 \\ \vdots \\ v_{m-1,x}^2 \\ v_{m-1,y}^2 \\ v_{m-1,z}^2 \end{matrix} \right]_m^T J \\
 &= \left( \|v_{0,n}^T\|_n^2 \dots \|v_{m-1,n}^T\|_n^2 \right)_{mn} \left( \begin{matrix} n J_{v_0,3n}^1 & & & \\ J_{v_0}^2 & J_{v_1}^2 & & \\ \vdots & \vdots & \ddots & \\ J_{v_0}^m & J_{v_1}^m & \dots & J_{v_{m-1}}^m \end{matrix} \right)_{3mn}
 \end{aligned}$$

⑥

$$\text{where } \left\| \mathbf{v}_i \right\|_n^2 = \begin{pmatrix} \left\| \mathbf{v}_{loc=1} \right\|^2 \\ \vdots \\ \left\| \mathbf{v}_{loc=n} \right\|^2 \end{pmatrix}$$

$$= \begin{pmatrix} D_1 & D_2 & \cdots & D_m \end{pmatrix}_{3mn}$$

$$\text{where } D_k = \sum_{i=k+1}^m \left\| \mathbf{v}_{i-1}^T \right\|_n^2 J_{\mathbf{v}_{k-1}}^i \quad k=1, \dots, m.$$

$$\begin{aligned} 2^\circ \quad \frac{\partial P_2(w)}{\partial w} &= \frac{1}{2} \frac{\partial}{\partial w} \left\| \rho_m - \rho_i^{obs} \right\|^2 = (\rho_m - \rho_i^{obs})^T \frac{\partial \rho_m}{\partial w}_{3mn} \\ &= (\rho_m - \rho_i^{obs})^T (J_{\mathbf{v}_0}^m J_{\mathbf{v}_1}^m \cdots J_{\mathbf{v}_{m-1}}^m). \end{aligned}$$

$$\begin{aligned} 3^\circ \quad \frac{\partial P_3(w)}{\partial w} &= \frac{1}{2} \text{hd} \cdot \text{dt} \frac{\partial}{\partial w} \left\| \text{Grad } w \right\|^2 = \text{hd} \cdot \text{dt} (\text{Grad } w)^T \text{Grad} \\ &= \text{hd} \cdot \text{dt} w^T \text{Grad}^T \text{Grad}. \end{aligned}$$

$$\text{For the Hessian matrix } H = \left( \frac{\partial^2 P}{\partial v_i \partial v_j} \right)_{i,j} = \frac{\partial g}{\partial w}_{3mn}$$

$$\begin{aligned} \text{recall that } g &= \beta \text{hd} \cdot \text{dt} \left[ 2 P^T M \text{diag}(w) + [M(w \otimes w)]^T J \right] \\ &\quad + (\rho_m - \rho_i^{obs})^T \frac{\partial \rho_m}{\partial w} + \gamma \cdot \text{hd} \cdot \text{dt} w^T \text{Grad}^T \text{Grad} \end{aligned}$$

(7)

$$\begin{aligned}
 \Rightarrow H = \frac{\partial g}{\partial v} &= \beta \cdot h d \cdot dt \left[ \underbrace{2p^T \nabla(M \text{diag}(w)) +}_{\text{red}} \right. \\
 &\quad \nabla(2p) M \text{diag}(w) + \\
 &\quad M(w \otimes w) \nabla J + \nabla[M(w \otimes w)] J \left. \right] \\
 &+ \underbrace{\left( \frac{\partial p_m}{\partial v} \right)^T \left( \frac{\partial p_m}{\partial v} \right)}_{\text{red}} + (p_m - p_{\text{obs}}) \frac{\partial^2 p_m}{\partial v^2} \\
 &+ \underbrace{\gamma \cdot h d \cdot dt \text{Grad}^T \cdot \text{Grad}}_{\text{red}}
 \end{aligned}$$

numerically we approximate  $H$  with only first three terms in red, so

$$\begin{aligned}
 H \approx 2\beta \cdot h d \cdot dt \cdot \text{diag} \left( \underbrace{p^T M}_{1 \text{mn} 3 \text{mn}} \right) + \underbrace{\left( \frac{\partial p_m}{\partial v} \right)^T \left( \frac{\partial p_m}{\partial v} \right)}_{3 \text{mn}} \\
 + \gamma \cdot h d \cdot dt \text{Grad}^T \cdot \text{Grad}
 \end{aligned}$$

Next we go to the code.

In driver.m, we have consecutive observed image  $p_0^{\text{obs}}, p_1^{\text{obs}}, \dots, p_{S-1}^{\text{obs}}$ .  
 but in our note, we only consider one loop:  $p_0^{\text{obs}} \rightarrow p^{\text{obs}}$   
 in the next loop, we use  $p_m \rightarrow p_2^{\text{obs}}$  where  $p_m$  is the last-time  
 image computed by DMT from the previous loop.

In paramInitFunc.m, we load in the parameters in the model,  
 as well as precompute some necessary matrices, and save in structure "par".

$$\text{par.n} = [N_x, N_y, N_z]$$

$$\text{par.h1} = \text{par.h2} = \text{par.h3} = 1.$$

$$\text{par.hd} = h_1 \cdot h_2 \cdot h_3 = hd = 1.$$

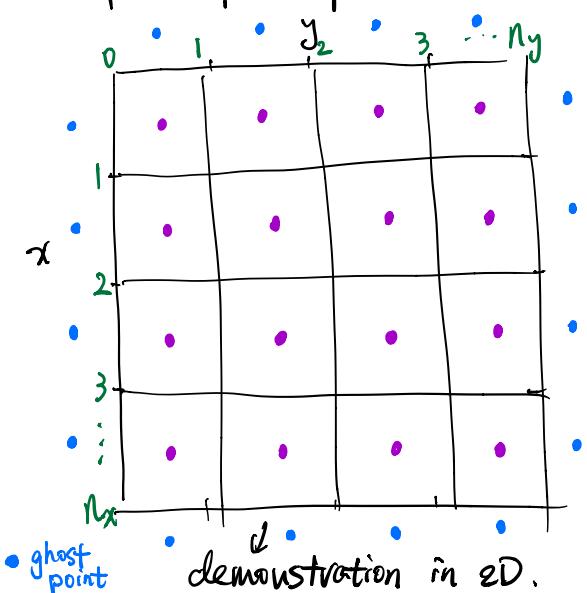
$$\text{par.dt} = dt \text{ in note}$$

$$\text{par.mt} = m \text{ in note}$$

$$\text{par.Sigma} = 6 \text{ in note}$$

We use boundary condition as cell-centered Neumann condition.

par.Xc, par.Yc, par.Zc are cell-centered grid.



par.Grad = gradient in space.

$$= \begin{pmatrix} G_x \\ G_y \\ G_z \end{pmatrix} \rightarrow \begin{array}{l} (N_x+1) N_y N_z \times N_x N_y N_z \\ N_x (N_y+1) N_z \times N_x N_y N_z \\ N_x N_y (N_z+1) \times N_x N_y N_z \end{array}$$

when using Neumann boundary,  
 we set ghost points on the spatial  
 boundary, such that the value equals  
 the nearest inside-neighbor.

(9)

In GN block-u.m,

$\rho_{0-i} = \rho_0 = \rho_0^{\text{obs}}$  in note.

$A_{\text{big}} = M$  in note.

$\phi_i = \text{get\_phi}(Rho\_i, u, nt, dt, par) \rightarrow \text{calculate } \bar{F}^{(n)}$ .  
energy functional.

$d_{\text{mk2}}$  = second term of  $\frac{\partial \bar{F}_i(u)}{\partial u}$ .

$u = v$  in note.

$\rho_0 = \text{advection}(Rho\_i, u, nt, dt, par) \rightarrow$  get  $\rho = \begin{pmatrix} \rho_1 \\ \vdots \\ \rho_m \end{pmatrix}$  from

$\rho_0$  and  $v$  following  
the adv-diff eq.

Then in a loop, we get  $d_{\text{mk2}}$  — second term of

$$\frac{\partial \bar{F}_i(u)}{\partial u}.$$

vertical  
→ vector

$$g = \beta \frac{\partial \bar{F}_1(u)}{\partial u} + \frac{\partial \bar{F}_2(u)}{\partial u} + \gamma \frac{\partial \bar{F}_3(u)}{\partial u}$$

$$= \beta \left[ 2 \text{hd} \cdot dt \rho^T M \text{diag}(v) + d_{\text{mk2}} \right]^T + \begin{pmatrix} J_{V_0}^m \\ J_{V_1}^m \\ \vdots \\ J_{V_{m-1}}^m \end{pmatrix} (\rho_m - \rho_0^{\text{obs}})$$

$$+ \gamma \text{hd} \cdot dt \cdot \text{Grad}^T \text{Grad } v$$

$$Hx = 2\beta \cdot h \cdot dt \cdot \underbrace{\text{diag}(P^T M)}_{3mn} x + \underbrace{\left( \frac{\partial P_m}{\partial v} \right)^T}_{n} \frac{\partial P_m}{\partial v} x, \quad \begin{matrix} \text{linear} \\ \text{form.} \end{matrix}$$

+  $\gamma \cdot h \cdot dt \cdot \text{Grad}^T \text{Grad} x$

$$\left( H = 2\beta \cdot h \cdot dt \cdot \text{diag}(P^T M) + \left( \frac{\partial P_m}{\partial v} \right)^T \frac{\partial P_m}{\partial v} \right) \begin{matrix} \text{matrix} \\ \text{form} \end{matrix}$$

+  $\gamma \cdot h \cdot dt \cdot \text{Grad}^T \text{Grad}$

$$\frac{\partial P_m}{\partial v} x \rightarrow \text{get\_drNdu}(x)$$

$$\left( \frac{\partial P_m}{\partial v} \right)^T \left( \frac{\partial P_m}{\partial v} x \right) \rightarrow \text{get\_drNdu}^T \left( \frac{\partial P_m}{\partial v} x \right)$$

Solve  $HS = -g$  for  $S$ .

linear search.

In advecDiff.m,

recall that, given  $\rho_0$  and  $v$ , we can derive  $\rho = \begin{pmatrix} \rho_0 \\ \vdots \\ \rho_m \end{pmatrix}$ ,  
with  $\left\{ \begin{array}{l} \rho_i^{\text{adv}} = S(v_i) \rho_i \\ (I_n - dt Q) \rho_{i+1} = \rho_i^{\text{adv}}, i=0, \dots, m-1. \end{array} \right.$

$M_{\text{dis}} = Q$  in note.

$$\rho_0 = \begin{pmatrix} \rho_0 \\ \vdots \\ \rho_m \end{pmatrix} = \begin{pmatrix} \rho_0 \\ \vdots \\ \rho_m \end{pmatrix} \text{ in note.}$$

		0	1	2	...	m	
in note		i	1	2	3	...	nt+1
t	i	1	2	3	...	nt+1	
in code		$\rho_0$	$\xrightarrow{v_0} \rho_1$	$\xrightarrow{v_1} \rho_2$	$\xrightarrow{v_2} \dots$	$\xrightarrow{v_{m-1}} \rho_m$	

$S = \text{dTrilinear}(\rho_0(:, i), \dots) \rightarrow \text{get } S(v_i) \text{ in note.}$

$\rho_0(:, i) = S \cdot \rho_0(:, i-1) \rightarrow \text{get } S(v_i) \rho_i = \rho_i^{\text{adv}}$  in note.

$\rho_0(:, i) = \text{pcg}\left(I - dt * M_{\text{dis}}, \rho_0(:, i)\right)$



solve linear system

$(I_n - dt Q) \rho_{i+1} = \rho_i^{\text{adv}}$  for  $\rho_{i+1}$ .

only return  $\begin{pmatrix} \rho_1 \\ \vdots \\ \rho_m \end{pmatrix}$ .

12

To obtain  $\text{dmk2} = \text{hd} \cdot \text{dt} \cdot [M(v \odot v)]^T J_{mn}^3$

recall that

$$M(v \odot v) = \begin{pmatrix} \|v_0^T\|^2 & \dots & \|v_{m-1}^T\|^2 \end{pmatrix}_{mn}$$

$$J = \begin{pmatrix} J_{v_0}^1 & & & \\ J_{v_0}^2 & J_{v_1}^2 & & \\ \vdots & \vdots & \ddots & \\ J_{v_0}^m & J_{v_1}^m & \dots & J_{v_{m-1}}^m \end{pmatrix}_{3mn} \triangleq \begin{pmatrix} \frac{\partial P_1}{\partial v_{3mn}} \\ \frac{\partial P_2}{\partial v_{3mn}} \\ \vdots \\ \frac{\partial P_m}{\partial v_{3mn}} \end{pmatrix}$$

$$\Rightarrow M(v \odot v) J \triangleq \begin{pmatrix} D_1 & D_2 & \dots & D_m \end{pmatrix}_{3mn}$$

$$\text{where } D_k = \sum_{i=k}^m \|v_{i-1}^T\|^2 J_{v_{k-1}}^i, k=1, \dots, m.$$

Thus we have the table

$$\left\{ \begin{array}{l} D_m = \|v_{m-1}^T\|^2 J_{v_{m-1}}^m \\ D_{m-1} = \|v_{m-1}^T\|^2 J_{v_{m-2}}^m + \|v_{m-2}^T\|^2 J_{v_{m-2}}^{m-1} \\ \vdots \\ D_2 = \|v_{m-1}^T\|^2 J_{v_1}^m + \|v_{m-2}^T\|^2 J_{v_1}^{m-1} + \dots + \|v_1^T\|^2 J_{v_1}^2 \\ D_1 = \|v_{m-1}^T\|^2 J_{v_0}^m + \|v_{m-2}^T\|^2 J_{v_0}^{m-1} + \dots + \|v_1^T\|^2 J_{v_0}^2 + \|v_0^T\|^2 J_{v_0}^1 \end{array} \right. \quad \begin{array}{c} E_m \\ E_{m-1} \\ \swarrow \\ E_2 \\ \leftarrow \\ E_1 \end{array} \quad \begin{array}{c} \uparrow \\ \uparrow \\ \uparrow \\ \uparrow \\ \uparrow \\ \uparrow \end{array}$$

(13)

As long as we can computer  $D_1, \dots, D_m$ .

even though  $(D_1, \dots, D_m)$  is of dimension  $(\times 3mn)$ .

it doesn't hurt to compute  $\tilde{D} = \begin{pmatrix} D_1 \\ \vdots \\ D_m \end{pmatrix}$  of dimension  $m \times 3n$ .

In observation of the special form of  $(*)$ ,

$$D = E_1 + E_2 + \dots + E_{m-1} + E_m$$

$$\text{where } E_i = \begin{pmatrix} \|v_{i1}^T\|^2 J_{v_0} \\ \vdots \\ \|v_{i1}^T\|^2 J_{v_{i-1}} \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \text{diag}(\|v_{i1}^T\|^2, m) \begin{pmatrix} J_{v_0} \\ \vdots \\ J_{v_{i-1}} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

In practice, only need to compute those non-zero entries.

$$\Rightarrow \left( E_i \right)_{\substack{i \\ 3n}}^{\text{first } i \text{ entries}} = \text{diag}(\|v_{i1}^T\|^2, i) \left( \frac{\partial p_i}{\partial v_{\text{first } i \text{ entries}}} \right)_{3n}^{\text{vertical order}}$$

We know that in our model,  $nt = m$ .  $v = \begin{pmatrix} v_0 \\ \vdots \\ v_{m-1} \end{pmatrix}$ .

~~A~~ to get  $\frac{\partial p_i}{\partial v_{\text{first } i \text{ entries}}}$ , we can assume in this case,

$nt = i$ , then  $\downarrow$  is the final density function's derivative wrt.  $v = \begin{pmatrix} v_0 \\ \vdots \\ v_{i-1} \end{pmatrix}$  with other parameters like "dt" unchanged.

14

Go back to the code,  
in math  $g$  is of size  $1 \times 3mn$ .

but numerically we prefer to put  $g$  as vertical vector,

so we instead compute  $g^T$ .

$$\text{Then } \text{dank2} = \text{hd} \cdot \text{dt} \begin{matrix} \mathbf{J}^T \\ \begin{matrix} 3mn & mn \end{matrix} \end{matrix} \left[ \mathbf{M}(\mathbf{v} \otimes \mathbf{u}) \right] = \begin{pmatrix} \mathbf{D}_1^T \\ \vdots \\ \mathbf{D}_m^T \end{pmatrix}$$

Similarly, we instead compute  $\begin{pmatrix} \mathbf{D}_1^T & \dots & \mathbf{D}_m^T \end{pmatrix} = \mathbf{D}^T$ .

$$\Rightarrow \mathbf{D}^T = \begin{matrix} \mathbf{E}_1^T \\ \vdots \\ \mathbf{E}_m^T \end{matrix} + \dots + \begin{matrix} \mathbf{E}_i^T \\ \vdots \\ \mathbf{E}_m^T \end{matrix}, \text{ where } (\mathbf{E}_i^T)_{\text{first } i \text{ entries (horizontal)}}$$

$$\begin{aligned} &= \left( \mathbf{J}_{v_0}^{i^T} \dots \mathbf{J}_{v_{i-1}}^{i^T} \right) \text{diag} \left( \| \mathbf{v}_{i+1}^{\wedge} \|^2, i \right) \\ &= \left( \mathbf{J}_{v_0}^{i^T} \| \mathbf{v}_{i+1}^{\wedge} \|^2, \dots, \mathbf{J}_{v_{i-1}}^{i^T} \| \mathbf{v}_{i+1}^{\wedge} \|^2 \right). \end{aligned}$$

$$\text{where } \mathbf{J}_{v_j}^k = \mathbf{L}^T \mathbf{S}(\mathbf{v}_{k+1}^{\wedge}) \dots \mathbf{L}^T \mathbf{S}(\mathbf{v}_{j+1}^{\wedge}) \mathbf{L}^T \mathbf{B}(\rho_j)$$

In each entry, it is a matrix times a vector.

iteratively

$$\boxed{\text{dank2}} = \boxed{\quad} + \boxed{\quad} + \dots + \boxed{\quad} + \boxed{\quad}$$

So for  $j=1:m$ .

$$dmk2(:, 1:j) = dmk2(:, 1:j) + \\ h \cdot dt \left( J_{v_0}^{j^T} \| v_{j-1} \|^2, \dots, J_{v_{j-1}}^{j^T} \| v_{j-1} \|^2 \right)$$

end

$$\text{where } \| v_{j-1} \|^2 = (I_n | I_n | I_n) \begin{pmatrix} v_{j-1,x}^2 \\ v_{j-1,y}^2 \\ v_{j-1,z}^2 \end{pmatrix} = A(v_{j-1} \odot v_{j-1}).$$

In get\_drdut.m

this function returns flattened

$$\text{SensTx} = \left( J_{v_0}^{m^T} y, \dots, J_{v_{m-1}}^{m^T} y \right)_m^T = \frac{(\partial P_m)^T}{3mn} y,$$

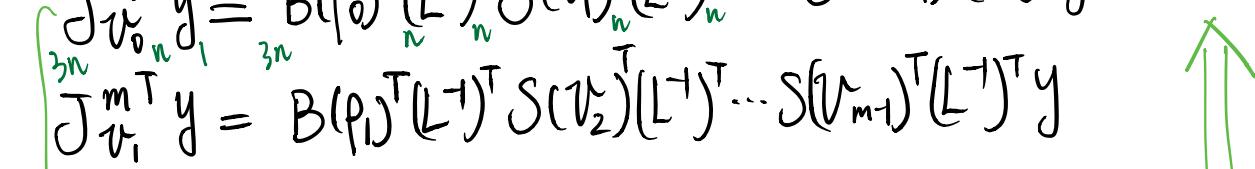
$$B = I_n - dt Q = L \text{ in note.}$$

$$Rho = (P_0 \ P_1 \ \dots \ P_m) \text{ in note.}$$

By the expression of  $J_{v_k}^m = L^{-1} S(v_{m-1}) \dots L^{-1} S(v_{k+1}) L^{-1} B(P_k)$

$$\text{where } B(P_k) = \frac{\partial}{\partial v_k} (S(v_k) P_k). \quad (\Delta)$$

we have table

$J_{v_0}^{m^T} y = B(P_0)^T (L^{-1})^T S(v_1)^T (L^{-1})^T \dots S(v_{m-1})^T (L^{-1})^T y$	$\vdots$ $\vdots$ $\vdots$ $\vdots$	
$J_{v_1}^{m^T} y = B(P_1)^T (L^{-1})^T S(v_2)^T (L^{-1})^T \dots S(v_{m-1})^T (L^{-1})^T y$		
$J_{v_{m-2}}^{m^T} y = B(P_{m-3})^T (L^{-1})^T S(v_{m-2})^T (L^{-1})^T \dots S(v_{m-1})^T (L^{-1})^T y$		
$J_{v_{m-1}}^{m^T} y = B(P_{m-1})^T (L^{-1})^T y$		

sens      sensI      sensI      sensI

(17)

It is obvious that as  $k$  increases, the expression will simplify. So it is better to compute from the end to the start.

Note that we can compute  $B(p_k)$  and  $S(v_k)$  at the same time in one function `dTrilinear.m`, which also can be seen from equation (4).

So start from  $J_{v_{m-1}}^{m^T} y$ , we can have both  $B(p_{m-1})^T (L^{-1})^T y$  and  $S(v_{m-1})^T (L^{-1})^T y$ , and the latter can be passed to the next loop as the vector.

$$dt \cdot [Tx, Ty, Tz] = B(p_k) \text{ in note.}$$

$$S = S(v_k) \text{ in note.}$$

Note that even though we write as  $L^{-1}$ , we are not computing the inverse, but by solving linear systems.

in dTriLinear.m,

this function returns averaging matrix  $P$  and derivatives  $T_x, T_y, T_z$ .

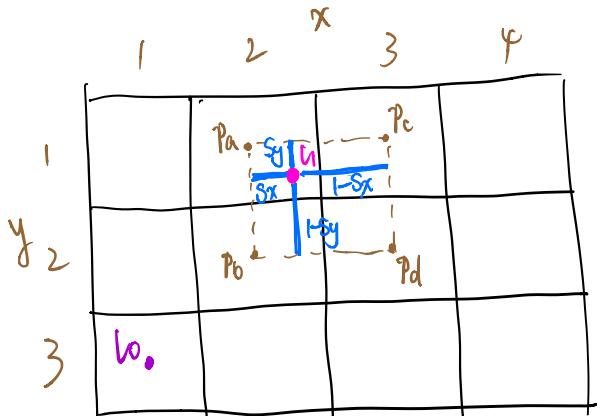
inputs:  $T - P_k$  in note

$x, y, z$  — new location at next time — old location +  $dt \cdot V_k^t$ .

outputs:  $P - S(V_k^t)$

$$dt [T_x, T_y, T_z] - B(P_k) = \frac{\partial}{\partial V_k^t} (S(V_k^t) P_k).$$

$P_k^{adv} = S(V_k^t) P_k$ . how to get expression of  $S(V_k^t)$ ?

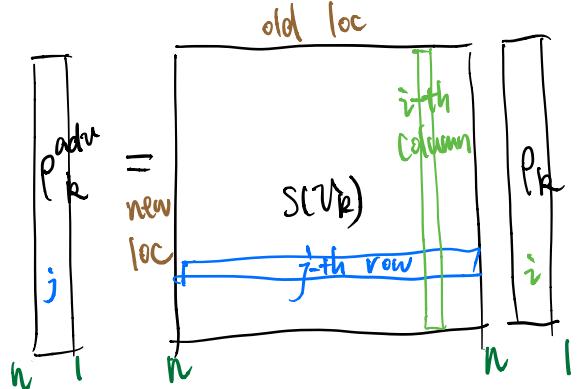


demonstrate in 2d:

$l_0$  is moved directly to  $l_1$ .  
mass at  $l_1$  can be split into 4 locations  $P_a, P_b, P_c, P_d$  at ratio  
 $(1-S_x)(1-S_y), (1-S_x)S_y,$   
 $S_x(1-S_y), S_xS_y$ , respectively.

- starting grid point  $(x_c, y_c, z_c), l_0$
- end point  $(x_c + dt \cdot V_{k,x}, y_c + dt \cdot V_{k,y}, z_c + dt \cdot V_{k,z})$
- neighbor grid points

$$\text{where } S_x = \frac{x_c + dt \cdot V_{k,x} - [x_c + dt \cdot V_{k,x}]}{dt \cdot V_{k,x} - [dt \cdot V_{k,x}]}, \dots$$



19

in order to get matrix  $S(V_k)$ . The row-dimension is the new location axis; the column-dimension is the old-location axis.

The  $i$ -th column of  $S(V_k)$  contains the ratio of splitting of old location at  $i$  into new locations.

e.g. loc  $i$  is split at loc  $j_1, \dots, j_4$  with ratio  $r_1, \dots, r_4$ .

$$\text{then } S(V_k)(i, j_1) = r_1,$$

$$S(V_k)(i, j_2) = r_2,$$

$$S(V_k)(i, j_3) = r_3,$$

$$S(V_k)(i, j_4) = r_4.$$

The  $j$ -th row of  $S(V_k)$  contains where the new location  $j$  comes from.

In conclusion,  $\underline{S(V_k)(j, i)}$  = the ratio of mass at new location  $j$  coming from old location  $i$ .

So  $S(V_k)$  is only w.r.t  $V_k$ , not including  $P_k$ .

Next we explore the expression of  $B(P_k) = \frac{\partial}{\partial V_k} (S(V_k) P_k)$

according to the expression of  $P_k$ , it has nothing to do with  $V_k$ , then the derivative is actually posed on  $S(V_k)$ .

First of all,  $\underline{B(P_k)} = [B_x(P_k), B_y(P_k), B_z(P_k)]$ .

where  $B_x(P_k) = \frac{\partial}{\partial V_{k,x}} (S(V_k) P_k), \dots, \dots$ .

We can just focus on  $Bx(P_k)$ , since the rest dimensions are straightforward if  $x$ -dimension is solved.

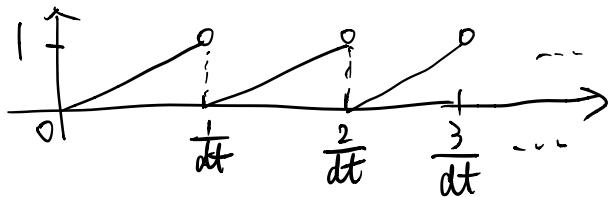
Take 2d as example, recall that the non-zero entries are  $(1-S_x)(1-S_y)$ ,  $S_x(1-S_y)$ ,  $(1-S_x)S_y$ ,  $S_x S_y$ .

$$\text{where } S_x = dt \cdot V_{k,x} - [dt \cdot V_{k,x}]$$

$$S_y = dt \cdot V_{k,y} - [dt \cdot V_{k,y}]$$

$$\Rightarrow \frac{\partial S_x}{\partial V_{k,x}} = dt, \quad \frac{\partial S_y}{\partial V_{k,x}} = 0.$$

Note, for function  $y(x) = dt \cdot x - [dt \cdot x]$



So  $y(x)$  is differentiable on  $\mathbb{R}/\mathbb{Z} \Rightarrow$  almost everywhere.

$$\Rightarrow y'(x) = dt, \text{ a.e.}$$

$$\Rightarrow \frac{\partial ((1-S_x)(1-S_y))}{\partial V_{k,x}} = -dt(1-S_y), \quad \frac{\partial (1-S_x)S_y}{\partial V_{k,x}} = -dt S_y$$

$$\frac{\partial S_x(1-S_y)}{\partial V_{k,x}} = dt(1-S_y), \quad \frac{\partial S_x S_y}{\partial V_{k,x}} = dt \cdot S_y.$$

$$\Rightarrow B_x(\rho_k) = \underbrace{T_x(v_k)}_n \underbrace{\text{diag}(\rho_k)}_n$$

where the entries of  $T_x(v_k)$  are the entry-wise derivatives w.r.t  $v_{k,x}$  of  $S(v_k)$ .

Similarly to get  $T_y(v_k), T_z(v_k)$ .

$$\Rightarrow B(\rho_k) = dt \cdot [T_x, T_y, T_z]$$

$\downarrow$      $\downarrow$      $\downarrow$   
 in code.

in `get_dvNdu.m`,

This function returns

$$\left( J_{V_0}^m, J_{V_1}^m, \dots, J_{V_{m-1}}^m \right) \underset{3n}{\overset{n}{x}} = \frac{\partial P_m}{\partial V} \underset{3n}{\overset{n}{x}}$$

If we write  $x = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix}$ , then the output is

$$(J_{V_0}^m, \dots, J_{V_{m-1}}^m) \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} = \sum_{i=1}^m J_{V_{i-1}}^m x_i$$

$B = I_n - dtQ = L$  in note.

$\text{rho} = (\rho_0, \rho_1, \dots, \rho_m)$  in note.

recall that  $J_{V_R}^m = L^{-1} S(V_{m-1}) \cdots L^{-1} S(V_{k+1}) L^{-1} B(P_k)$ .

$$\text{where } B(P_k) = \frac{\partial}{\partial V_k} (S(V_k) P_k).$$

Then we have the table:

$$\left\{ \begin{array}{l} J_{V_0}^m x_1 = L^{-1} \underbrace{S(V_{m-1})}_{\text{mth}} L^{-1} \underbrace{S(V_{m-2})}_{\text{3rd}} \cdots L^{-1} \underbrace{S(V_2)}_{\text{2nd}} L^{-1} \underbrace{S(V_1)}_{\text{2nd}} L^{-1} \underbrace{B(P_0)}_{\text{3rd}} x_1 \\ J_{V_1}^m x_2 = L^{-1} \underbrace{S(V_{m-1})}_{\text{mth}} L^{-1} \underbrace{S(V_{m-2})}_{\text{3rd}} \cdots L^{-1} \underbrace{S(V_2)}_{\text{2nd}} L^{-1} \underbrace{B(P_1)}_{\text{3rd}} x_2 \\ \vdots \\ J_{V_{m-2}}^m x_{m-1} = L^{-1} \underbrace{S(V_{m-1})}_{\text{mth}} L^{-1} \underbrace{B(P_{m-2})}_{\text{2nd}} x_{m-1}, \\ J_{V_{m-1}}^m x_m = L^{-1} \underbrace{B(P_{m-1})}_{\text{2nd}} x_m \end{array} \right.$$

(23)

Calculating  $B(p_k)$  and  $S(p_k)$  can be done in  
the  $(k+1)$ -th loop.

So loop from 1 to  $m$ .