Table of Contents

add tools

```
cd ../functions
addpath(genpath(pwd))
cd ..
cd ./task15l1migration/
addpath(genpath('/Volumes/Users/linamiao/Dropbox/PQN/pqnl1'))
% addpath(genpath('e:\research\Tools\tristan'))
% addpath(genpath('e:\research\Tools\spot-slim'))
% addpath(genpath('e:\research\Tools\pSPOT'))
```

```
Error using cd
Cannot CD to ../functions (Name is nonexistent or not a directory).

Error in script (line 2)
cd ../functions
```

generate data

define model

```
model.o = [0 0];
model.d = [10 10];
model.n = [101 101];
model.nb = [30 30 0];
model.t0 = -.08;

t    = 0:.004:2.044; nt = length(t);
freq = fkk(t); nf = length(freq);
%freq = 0:1/t(end):.5/(t(2)-t(1)); nf = length(freq);
If   = [10:5:40];

model.freq = freq(If);            nfreq = length(model.freq);
model.zsrc = 10;
model.xsrc = linspace(0,1000,21);  nsrc  = length(model.xsrc);
model.zrec = 10;
model.xrec = linspace(0,1000,101); nrec  = length(model.xrec);
model.f0   = 10;
```

```
model.t0   = -.08;

xr = model.xrec; nr = length(xr);
xs = model.xsrc; ns = length(xs);

Q = speye(nsrc);

z = 0:10:1000;
x = 0:10:1000;
[zz,xx] = ndgrid(z,x);

% background velocity [m/s]
v0 = 2000 + 0.*zz;
dv = 0*xx;
dv(zz >= 520) = 100;
dv(zz >= 560) = 0;
figure; imagesc(dv);
v = v0+dv;
m0 = 1e6./v0(:).^2;
m = 1e6./v(:).^2;
dm = m-m0;
```

data

generate data

```
[~,J] = F(m0,Q,model);

% linearize data
Dobs = J*dm;
Dobs = gather(Dobs);

save temp Dobs
load temp
% permute into f-x-x order
cube = permute(reshape((Dobs),nrec,nsrc,length(model.freq)),[3 1 2]);

temp = zeros(nf,nrec,nsrc);
temp(If,:,:) = cube([1:length(model.freq)],:,:);
CUBE = temp;

% window data
CUBE = CUBE(:,1:5:end,:);
xr = xs;
nr = length(xr);
```

frquency domain migration

compensate negative frequency if mod(nf,2) == 1 tmp = [CUBE ; conj(CUBE(end:-1:2,:,:))]; ff = [freq, conj(freq(end:-1:2))]; else tmp = [CUBE ; conj(CUBE(end-1:-1:2,:,:))]; ff = [freq, conj(freq(end-1:-1:2))]; end CUBE = tmp;

```
op = opDSR_mig_freq(size(CUBE),freq,xr,xs,z,2000*ones(length(z),1),0);
```

```
af = op*vec(CUBE);
af = reshape(af,length(z),length(xr));
figure;imagesc(real(af));title('freq domain migration')
```

time domain migration

apply an inverse fourier transform to the CUBE

```
Dt = ifft(CUBE,[],1)*sqrt(size(CUBE,1));

figure;imagesc(real(Dt(:,:,11)));
op = opDSR_mig_time(size(Dt),t,xr,xs,z,2000*ones(length(z),1),0);
a = op*vec(Dt);
a = reshape(a,length(z),length(xr));
figure ;imagesc(real(a));title('time domain migration')
```

migration

```
    dim = size(CUBE);
    % frequency domain
    op3 = opDSR_mig_freq(dim,t,xr,xs,z,v,0);
    adjoint_test(op3)
    % time domain
    op4 = opDSR_mig_time(dim,t,xr,xs,z,v,0);
    adjoint_test(op4)
```

least square migration

```
temp = reshape(dm,model.n);
temp = temp(:,1:5:end);
temp = vec(temp);

% in freq domain
dim = size(CUBE);
A = opDSR_mig_freq(dim,freq,xr,xs,z,2000*ones(length(z),1),0);
B = A'*temp;
[X] = lsqr(A',B,[],10);
lsx = reshape(X,model.n(1),length(xr));
figure;imagesc(real(lsx));title('freq domain least square migration')

% in time domain
dim = size(Dt);
At = opDSR_mig_time(dim,t,xr,xs,z,2000*ones(length(z),1),0);
Bt = At'*temp;
[Xt] = lsqr(At',Bt,[],10);
lsxt = reshape(Xt,model.n(1),length(xr));
figure;imagesc(real(lsxt));title('time domain least square migration')
```

l1 migration

sparse transform % wavelet operator along time axis W = opSplineWavelet(nt, 1, nt, 3, 5);

% curvelet operator along source-receiver oordinates C = opCurvelet(nr,ns,6,16,1,'ME',0);

% oppKron2Lo : kronecker tensor product to act on a distributed vector S = oppKron2Lo(C, W', 1);

```
C = opCurvelet(length(z),nr,6,16,1,'ME',0);
aa = C*temp;
aa = sort(abs(aa),'descend');
figure; plot(aa);title('curvelet cooeficient')

% frequency domain
A = A'*C';
b = B;
opts.iteration = 200;
tt = C*temp;

% l1 migration
%[x_spg,r_spg,g_spg,info_spg] = spgl1(A, b, 0, 1e-3, zeros(size(tt)), opts);
[x_pqn,r_pqn,g_pqn,info_pqn] = pqnl1_2(A, b, 0, 1e-3, zeros(size(tt)), opts);



% time domain
At = At'*C';
b = Bt;
[xt_spg,rt_spg,gt_spg,infot_spg] = spgl1(A, b, 0, 1e-3, zeros(size(tt)), opts);
[xt_pqn,rt_pqn,gt_pqn,infot_pqn] = pqnl1_2(A, b, 0, 1e-3, zeros(size(tt)), opts);
```

Published with MATLAB® 7.13