

---

```

function image = DSR_mig(data,t,x,y,z,v)

% x is xr -- receiver grid
% y is xs -- source grid
% t      -- time axis
% z      -- vertical axis
% v      -- velocity as a function of z, must be the same size of z

if norm(x-y)
    error('source and receiver grids must be the same!');
end

if length(v) - length(z)
    error('velocity must be the same size as z')
end

% initialize image
image = zeros(length(z),length(x));

% depth step
dz = z(2) - z(1);

% basic computation of frequencies and wave numbers
dt = t(2)-t(1);
nt = length(t);
dx = x(2) - x(1);
nx = length(x);
dy = y(2) - y(1);
ny = length(y);
fny = .5/dt;
if mod(nt,2) == 1
    df = 2*fny/(nt-1);
    f = [0:df:fny -fny:df:-df];
else
    df = 2*fny/nt;
    f = [0:df:fny -fny+df:df:-df];
end
fny = .5/dx;
if mod(nx,2) == 1
    df = 2*fny/(nx-1);
    kx = [0:df:fny -fny:df:-df];
else
    df = 2*fny/nx;
    kx = [0:df:fny -fny+df:df:-df];
end
fny = .5/dy;
if mod(ny,2) == 1
    df = 2*fny/(ny-1);
    ky = [0:df:fny -fny:df:-df];
else
    df = 2*fny/ny;
    ky = [0:df:fny -fny+df:df:-df];

```

---

---

```

end

% f-k-k grid
[ff,kkx,kky] = ndgrid(f,kx,ky);

% loop over depth levels
for iz = 1:length(z)
    % survey sinking
    E = DSR_step(data,ff,kkx,kky,iz*dz,v(iz));

    % imaging condition
    % image(iz,:) = diag(squeeze(E(1,:,:)));
    image(iz,:) = diag(squeeze(sum(E,1)));

    % convert to midpoint-offset domain
    [F,y,h] = mh_convert(E,kkx,kky);

    % common angle imaging conditions
    cig = cig_imaging(F,y,h);
end

% end loop over depth levels

function v = DSR_step(u,ff,kkx,kky,dz,v)

% f-k-k transform
spec = fft(u,[],1);
spec = fft(spec,[],2);
spec = fft(spec,[],3);

% DSR operators
Px = 2*pi*sqrt((ff/v).^2-kkx.^2);
Py = 2*pi*sqrt((ff/v).^2-kky.^2);

Px = real(Px)+1i*abs(imag(Px));
Py = real(Py)+1i*abs(imag(Py));

% apply phase shift
spec = exp(1i*abs(dz)*(Px + Py)).*spec;

% inverse kk transform
% v = ifft(spec,[],1);
v = spec;
v = ifft(v,[],2);
v = ifft(v,[],3);

% take real part
v = real(v);

function [f,y,h] = mh_convert(e,kx,ky)
% this function is to convert wavefield from shot offset domain to midpoint
% offset domain, according to the formulation 24 25 given at:

```

---

---

```
% http://sepwww.stanford.edu/sep/prof/bei/sg/paper\_html/node12.html#SECTION0013400
% ks = .5*(ky-kh);
% kg = .5*(ky+kh);
```

```
function cig = cig_imaging(F,y,h)
ph = kh/f;
phmin = min(ph);
phmax = max(ph);
dph =
p = phmin:dph:phmax;
cig(km,pp,z) = sum();
```

```
Error: File: /Volumes/Users/linamiao/Dropbox/extended_migration/extended_m
Expression or statement is incomplete or incorrect.
```

*Published with MATLAB® 7.13*