Sampling Layered Models with the Metropolis Algorithm
Bill Menke, Nov. 21, 2015

Model function: Piecewise constant function $v(x)$ on an interval $(0, x_{max})$ with $M$ layers, motivated by a discussion of Bodin T., Yuan H., Romanowicz B., Inversion of receiver functions without deconvolution-application to the Indian craton. Geophys. J. Int. 2014;196:1025-1033.

Relationship between model and data: The data equals the model; I have stripped away the original receiver function application.

Probability of error: Normal distribution of the error $E = \|v^{true}(x) - v^{pre}(x)\|^2$. The variance $\sigma^2$ is prescribed.

Probability of prior information: Normal distribution of the squared difference between the number of layers $M$ and a prior value $L_a$. The prior value $L_a$ and variance $\sigma_a^2$ is prescribed.

Generating a proposed model from the current model:

With $\alpha \ll 1$ probability, the number of layers is decreased by unity. A randomly-chosen (via a uniform distribution) layer is merged with the one adjacent to it (and with greater $x$). The new layer inherits the average of the $v$'s of the two original layers.

With $\beta \leq \alpha$ probability, the number of layers is increased by unity. A randomly chosen layer is split in two, at a randomly chosen value of $x$ (uniform distribution). The two layers inherit the $v$ of the original.

The thickness and value of the successor model is then randomly perturbed (each by a Normal distribution with a mean of the current value and a prescribed variance). A layer is never allowed to be thinner than $x_{max}/100$. The total thickness is rescaled to $x_{max}$.

Case 1: True model is piecewise constant.

```
sigma = 0.07; Na = 2; sigmaN = 4; Niter=100000;
```

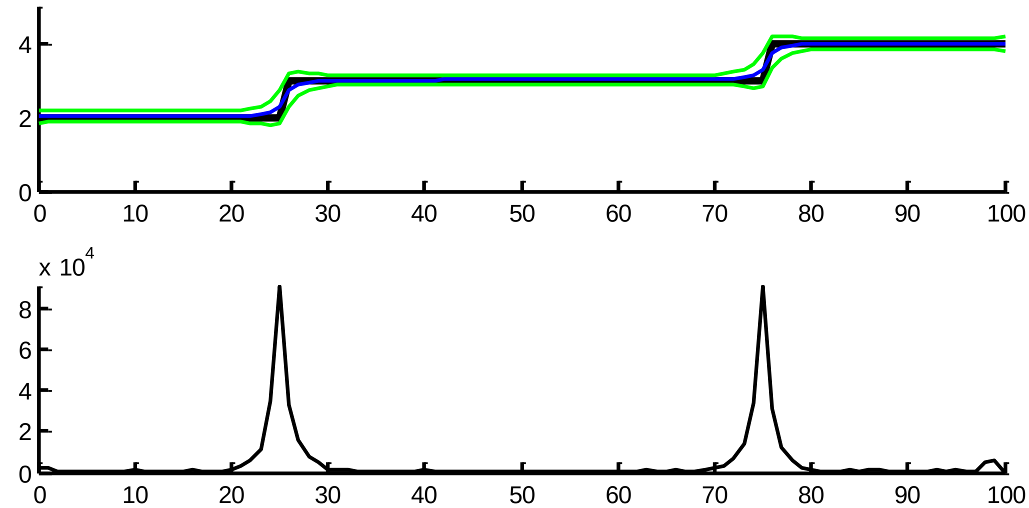


Figure 1. (top) True model (black), mean model (blue) and ± one standard deviation (green) (bottom) probability of interface.

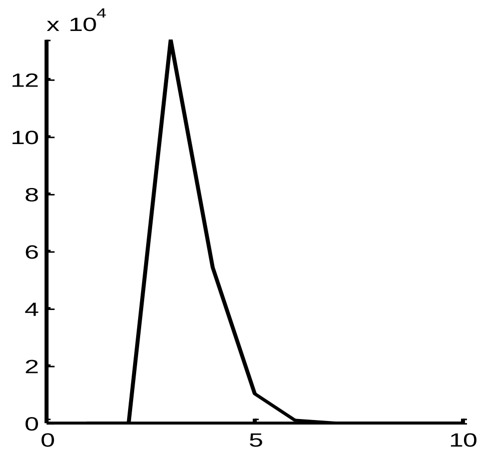


Figure 2. Probability of models with a given number of layers.

Case 2: True model is continuously varying.

```
sigma = 0.07; Na = 2; sigmaN = 4; Niter=100000;
```

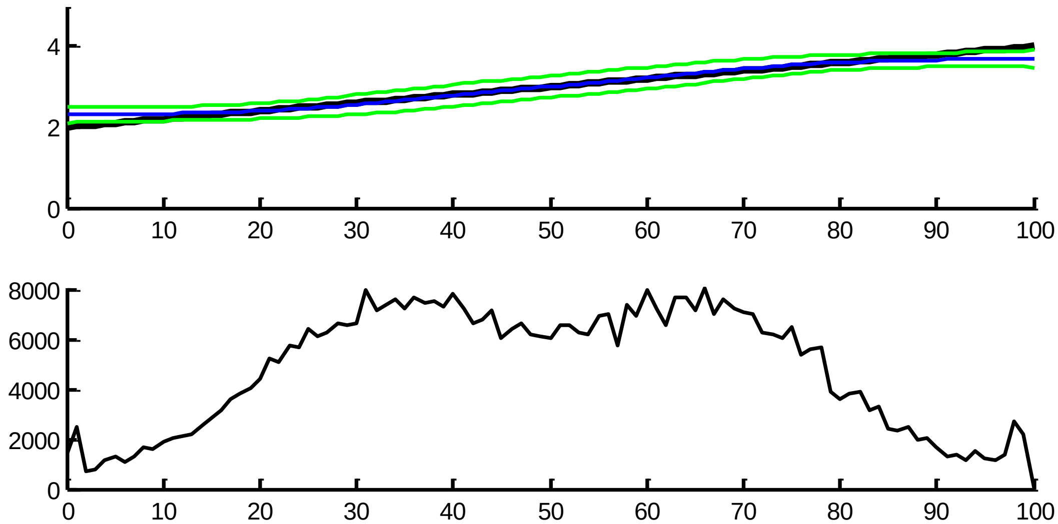


Figure 3. (top) True model (black), mean model (blue) and ± one standard deviation (green) (bottom) probability of interface.

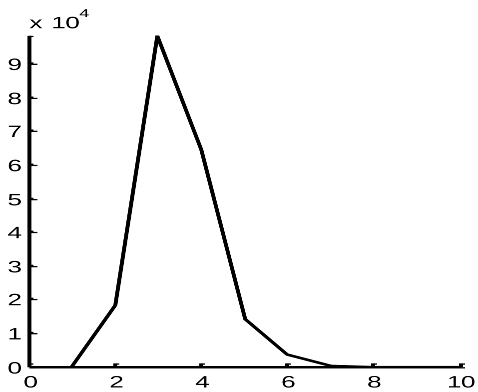


Figure 4. Probability of models with a given number of layers.

Question: For models with $N > 2$, is the probability of an interface larger at the center of the interval that at the ends, even when the layer thickness are random? In analogy to the binomial distribution, I would think that there are more combinations of layers thicknesses that put some layer interface near the center of the model than there are combinations that put an interface near an end.

```
clear all;
global N dx x

Niter=100000;

N=101;
Dx = 1.0;
x = Dx*[0:N-1]';
xmin=x(1);
xmax=x(N);

% state
strue.N = 3;
strue.z = [25,75,100]';
strue.v = [2,3,4]';

if( 1 )
    ptrue = stop(strue);
else
    ptrue = 2+2*x/xmax;
end
figure(1);
clf;
subplot(2,1,1);
set(gca,'LineWidth',2);
hold on;
axis( [xmin, xmax, 0, 5] );
plot(x,ptrue, 'k-', 'LineWidth', 3);

s0.N = 4;
s0.z = [15,65,80,100]';
s0.v = [1.5,2.5,3.1,3.9]';
p0 = stop(s0);

%plot(x,p0, 'r-', 'LineWidth', 2);
sigma = 0.07;
Na = 2;
sigmaN = 4;

sigma2 = sigma^2;
f = 1/( sqrt(2*pi) * sigma );
E0 = (ptrue-p0)'*(ptrue-p0)/N;
pE0 = f*exp( -0.5*E0/sigma2 );
sigma2N = sigmaN^2;
fN = 1/( sqrt(2*pi) * sigmaN );
pA0 = fN*exp( -0.5*((s0.N-Na)^2)/sigma2N);
pE0 = pE0*pA0;

psum=zeros(N,1);
```

```
p2sum=zeros(N,1);
iface = zeros(N,1);

s = s0;
p = p0;
E = E0;
pE = pE0;
Nr=0;
accepted=0;
for iter = [1:Niter]
    [sp, beta] = news( s );
    pp = stop( sp );
    Ep = (ptrue-pp)'*(ptrue-pp)/N;
    pEp = f*exp( -0.5*Ep/sigma2 );
    pA = exp( -0.5*((sp.N-Na)^2)/sigma2N);
    pEp = pEp * pA;
    r = beta*pEp/pE;
    if( r>1 )
        s = sp;
        p = pp;
        E = Ep;
        pE = pEp;
        accepted=accepted+1;
    else
        alpha = random('Uniform',0,1,1,1);
        if( r > alpha )
            s = sp;
            p = pp;
            E = Ep;
            pE = pEp;
            accepted=accepted+1;
        end
    end
    if( iter>1000 )
        Nr=Nr+1;
        psum=psum+p;
        p2sum = p2sum+p.^2;
        layers(Nr)=s.N;
        for ii=[1:s.N-1]
            zii = s.z(ii);
            jj = floor(zii/Dx) + 1;
            if( jj<1 )
                jj=1;
            elseif( jj>N )
                jj=N;
            end
            iface(jj)=iface(jj)+1;
        end
    end
```

```
end

pmean = psum/Nr;
psigma = sqrt((p2sum/Nr)-(psum/Nr).^2);
plot(x,pmean, 'b-', 'LineWidth', 2);
plot(x,pmean-psigma, 'g-', 'LineWidth', 2);
plot(x,pmean+psigma, 'g-', 'LineWidth', 2);

subplot(2,1,2);
set(gca,'LineWidth',2);
hold on;
axis( [xmin, xmax, 0, max(iface)] );
plot(x,iface, 'k-', 'LineWidth', 2);
fprintf('accepted models %d\n', accepted);

figure(2);
clf;
set(gca,'LineWidth',2);
hold on;
Lmax = 10;
layercount=zeros(Lmax,1);
for i=[1:Lmax]
    layercount(i) = length(find(layers==i));
end
axis( [0, Lmax, 0, max(layercount)] );
plot([1:Lmax],layercount, 'k-', 'LineWidth', 2);
```

```
function [ s2, beta ] = news( s1 )
global N dx x

s2 = s1;
M = s1.N;
zmax = s1.z(M);
t = random('Uniform',0,1,1,1);

sf = 3;

beta = 1;

% change number of layers
Pt = 0.1;
if( t < Pt )
    if( t < (Pt/sf) ) % subdivide layer
        beta = (1/sf)/(1-(1/sf));
        s2.v = zeros(M+1,1);
        s2.z = zeros(M+1,1);
```

```
        j = unidrnd(M);
        s2.N = M+1;
        s2.v(1:j) = s1.v(1:j);
        s2.v(j+1) = s1.v(j);
        s2.v(j+2:M+1) = s1.v(j+1:M);
        z1 = [0; s1.z];
        h1 = z1(2:M+1)-z1(1:M);
        h2 = zeros(M+1,1);
        h2(1:j-1) = h1(1:j-1);
        f = random('Uniform',0,1,1,1);
        h2(j) = f*h1(j);
        h2(j+1) = (1-f)*h1(j);
        h2(j+2:M+1) = h1(j+1:M);
        z2 = cumsum(h2);
        s2.z = z2;
    elseif (M>1) % coalesce layer
        beta = (1-(1/sf))/(1/sf);
        j = unidrnd(M-1);
        s2.v = zeros(M-1,1);
        s2.z = zeros(M-1,1);
        s2.N = M-1;
        s2.v(1:j-1) = s1.v(1:j-1);
        s2.v(j) = 0.5*(s1.v(j)+s1.v(j+1));
        s2.v(j+1:M-1) = s1.v(j+2:M);
        z1 = [0; s1.z];
        h1 = z1(2:M+1)-z1(1:M);
        h2 = zeros(M-1,1);
        h2(1:j-1) = h1(1:j-1);
        h2(j) = h1(j)+h1(j+1);
        h2(j+1:M-1) = h1(j+2:M);
        z2 = cumsum(h2);
        s2.z = z2;
    end
end

s1 = s2;
M = s1.N;
zmax = s1.z(M);

sigmav = 0.3;
sigmaz = 5;

s2.v = s1.v + random('Normal',0,sigmav,M,1);
z1 = [0; s1.z];
h1 = z1(2:M+1)-z1(1:M);
h2 = h1 + random('Normal',0,sigmaz,M,1);
h2(find(h2<1))=1;
H = sum(h2);
z2 = zmax*cumsum(h2)/H;
```

```
s2.z = z2;

end

function [ p ] = stop( s )
global N dx x
p = zeros(N,1);

j=1;
for i=[1:N];
    if( (x(i)>s.z(j)) && (j<s.N) )
        j=j+1;
    end
    p(i) = s.v(j);
end

end
```