



MATLAB Code:

```
function [Alpha,Beta,Gama]=model(depth,N_measured)

%The function receives two arrays as Input. The first is the pre-fall
depth of each
%Sampling point [m] and the second is the measured TCN concentration of
each sample [atoms/gr-qz].

% The function, using a stochastic procedure, returns 3 variables:
% alpha, Beta, and Gama. Alpha stand for 't' in Eq (1).

% All samples entered to the model run share an identical exposure age.
% Samples must be entered at increasing depth order. (the sample
closest to
% the surface will be the first)

%%Constants:
%10Be decay constant:
decay_constant=4.997456240518711e-07; % [1/year], Balco Cronus, V.2.2.1
%Rock density:
rock_density=2.2; %[gr/cm^3], measured
%Attenuation coefficient:
attenuation=160; %[gr/cm^-2], Balco Cronus, V.2.2.1
%Absorption rate calculation:
absorption_rate=rock_density/attenuation; %[1/cm]
%Production rate at site:
P=3.44; %[atoms/gr/year]. Calculated using Sea Level High Latitude
production rate of: 4.43±0.52, Balco Cronus, V.2.2.1,
%Using Dunai (2000,2001) for scaling, and Korte and Constable, (2005)
and
```

```

%Korte et al. (2005) to account for geomagnetic fluctuation.

%Initial limits for alpha and beta:

AlphaMax=10000;

BetaMax=25000;

%%Algorithm:

%Number of entered samples:

n=length(depth);

%Setting initial distribution limits for Beta, Gama, and Alpha:

minv=zeros(1,n*2+1);

maxv=[repmat(BetaMax,1,n),repmat(1,1,n),AlphaMax];

cells=1000000; % Size of random vector

for nn=1:5

%Determine random values for Beta using uniform distributions:

%Values of Beta are forced to decrease with depth, in the case of two
samples

%with identical pre-fall depth a 10% variation is allowed:

count_identical_depth=0;

for i=1:n

    if i==1

        Beta(i,:)=random('unif',minv(1,i),maxv(1,i),[1,cells]);

    else

        if depth(i-1)<depth(i)

            Beta(i,:)=random('unif',minv(1,i),Beta(i-1-
count_identical_depth,:),[1,cells]);

            count_identical_depth=0;

```

```

else
    count_identical_depth=count_identical_depth+1;
    Beta(i,:)=random('unif',Beta(i-1,:)-0.1.*Beta(i-
1,:),Beta(i-1,:)+0.1.*Beta(i-1,:),[1,cells]);
end
end
end
%Determine random values for Gama using uniform distributions:
for i=1:n
    Gama(i,:)=random('unif',minv(i+n),maxv(i+n),[1,cells]);
end
%Determine random values for Alpha using uniform distributions:
Alpha= repmat(random('unif',minv(2*n+1),maxv(2*n+1),[1,cells]),n,1);
%Calculating TCN concentration following Lal, (1991)
%Samples are assumed to be at zero depth since exposure
%Erosion is assumed to be zero
N_calc=Beta.*exp(-decay_constant.*Alpha)+P.*Gama.*exp(-
absorption_rate.*0).*(1-exp(-decay_constant.*Alpha))./decay_constant;
N=repmat((N_measured)',1,cells);
%RMSD and Bias functions:
RMSD=sqrt(1./n.*sum((N-N_calc).^2));
BIAS=abs(1/n.*sum(N)-1/n.*sum(N_calc));
%Setting a limit for RMSD values
R1=quantile(RMSD,0.01);
index=find(RMSD<R1);
%Choosing result according to the defined limit:
result=[Beta(:,index)' Gama(:,index)' Alpha(1,index)' RMSD(index)'
BIAS(index)'];
%Setting new distribution limits for the next iteration

```

```

for i=1:2*n+1
    minv(i)=quantile(result(:,i),0.25);
    maxv(i)=quantile(result(:,i),0.75);
end
end
% Calculate Pareto solutions
flag=1;
while (flag==1 && size(result,1)>=2)
    flag=0;
    for k=1:size(result,1)-1
        for i=k+1:size(result,1)
            if (result(k,2*n+2)>result(i,2*n+2) &&
result(k,2*n+3)>result(i,2*n+3))
                flag=1;
                result(k,:)=[];
            elseif (result(k,2*n+2)<result(i,2*n+2) &&
result(k,2*n+3)<result(i,2*n+3))
                flag=1;
                result(i,:)=[];
            end
            if (flag)
                break
            end
        end
    end
    if (flag)
        break
    end
end
end
end
end

```

```
%Final results  
  
Beta=mean(result(:,1:n));  
  
Gama=mean(result(:,n+1:2*n));  
  
Alpha=mean(result(:,2*n+1));  
  
end
```