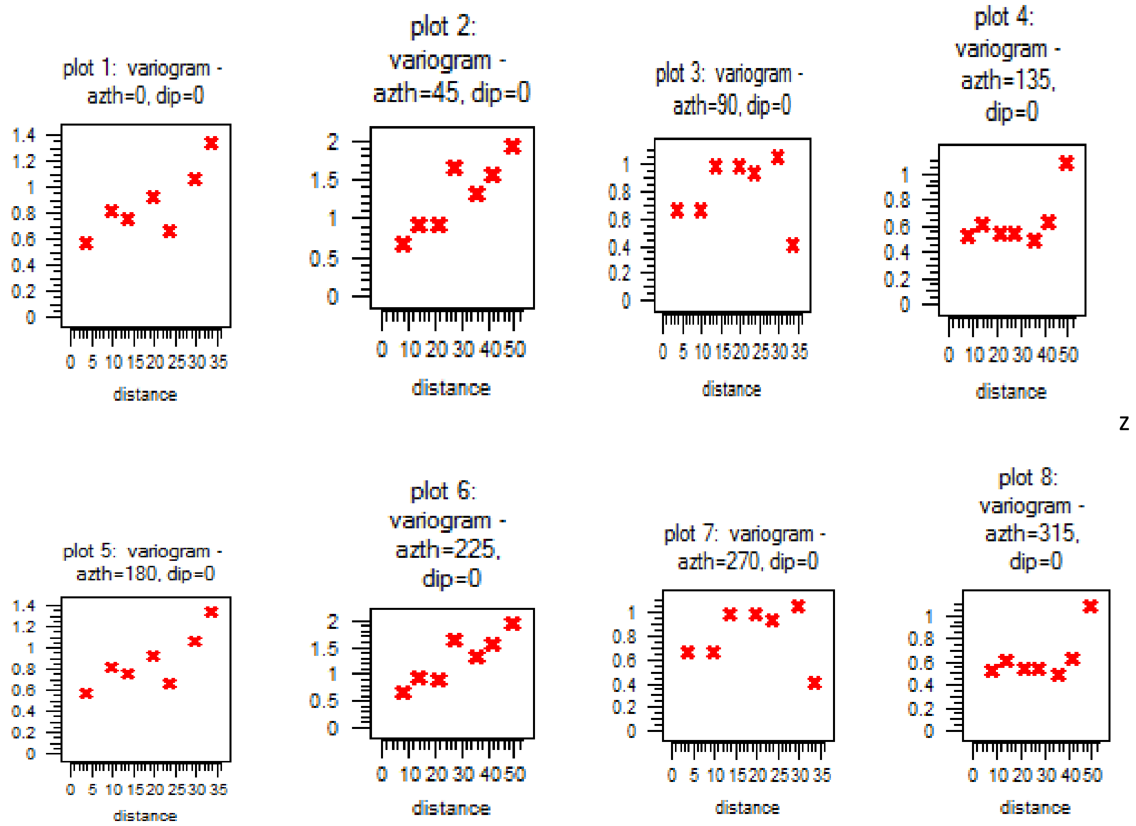
PROJECT-2 (REPORT)

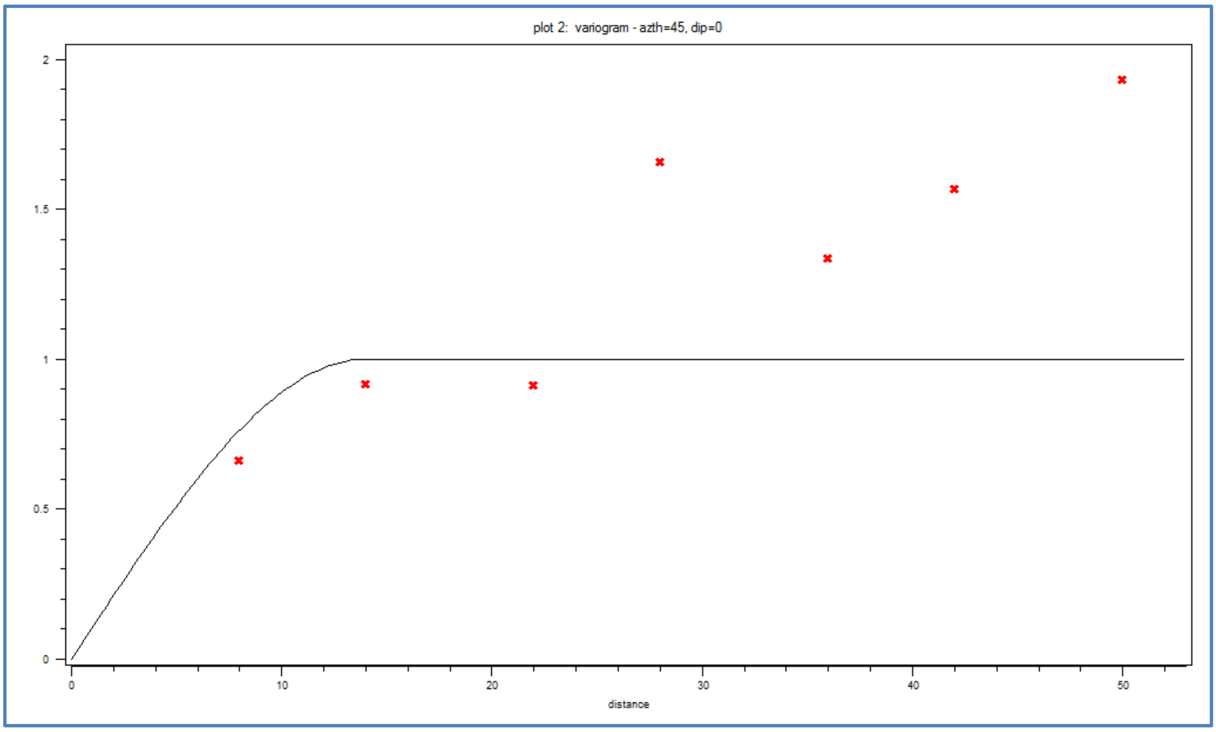
Q.1. **Variogram Modeling in SGEMS**

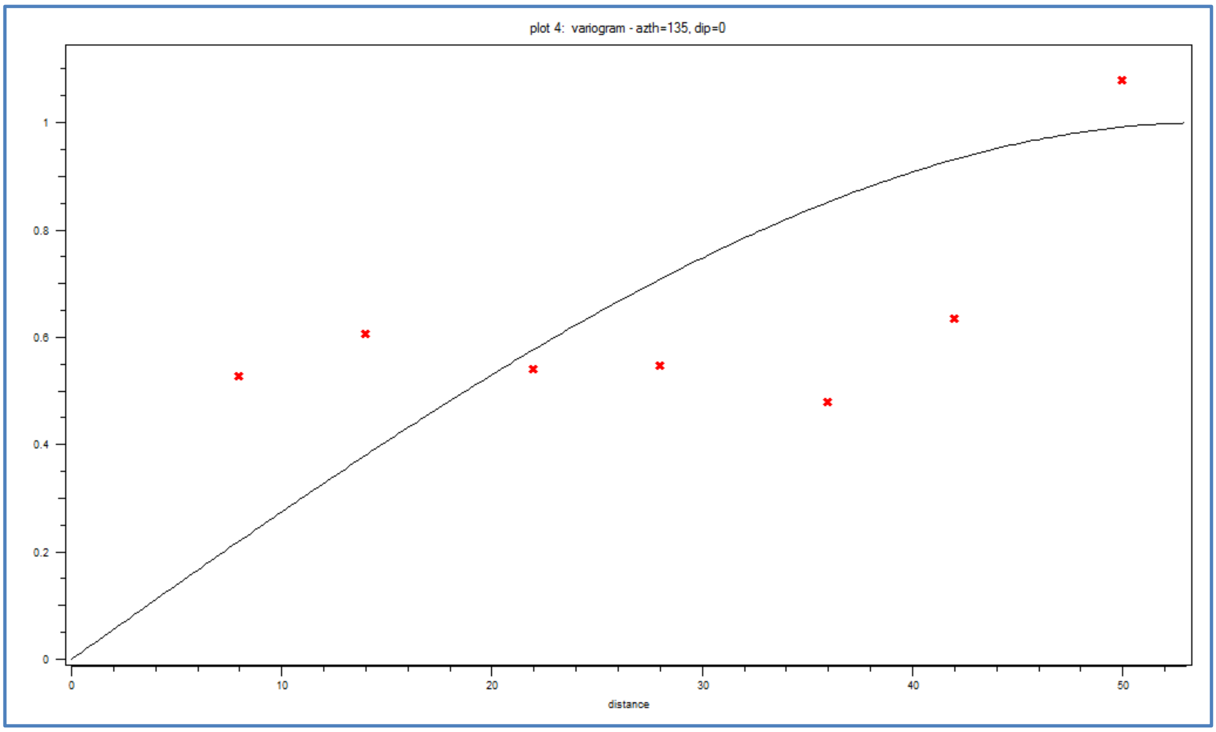


The variogram models shown are for the normal score transformed data of log(Permeability). The 135 and 315 variograms show a sill has been reached but it is not 1. Clearly, there is an anisotropy in the azimuth=135 degree direction. The anisotropy in the 135 degree can also be seen in the true data. The azimuth=135 will be the major axis of the anisotropy and the perpendicular direction, i.e. azimuth=45 will be the minor axis of anisotropy.

Q. 2.

The variogram model chosen is **spherical** since the property modelled is permeability. Permeability changes sharply and spherical variogram is suitable for modelling this sharp change.





The azimuth=45 variogram reaches a sill at around 14m. The azimuth-135 never reaches the sill=1. So, I have taken a range of 56 m (maximum distance in the model) for the 135 direction.

Major axis: Azimuth=135, Range=56 m

Minor Axis: Azimuth=45, Range=14 m

Variogram –Spherical (Permeability varies sharply and spherical variogram varies sharply as well)

Axes- has been rotated in the counterclockwise-direction by 45 degrees.

$$h = sqrt((\frac{hx}{ax})^2 + (\frac{hy}{ay})^2)$$

$$hx - distance\ along\ the\ transformed\ x - axis$$

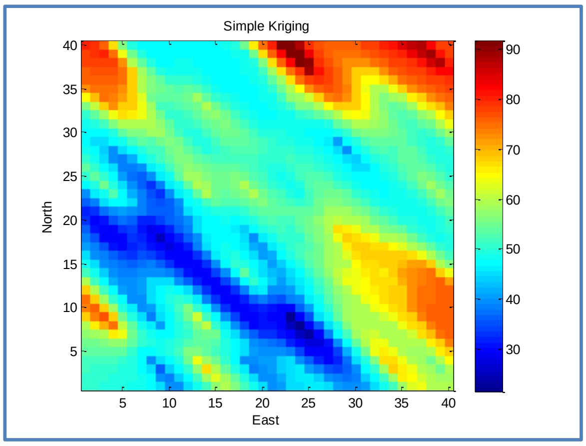
$$hy - distance\ along\ the\ transformed\ y - axis$$

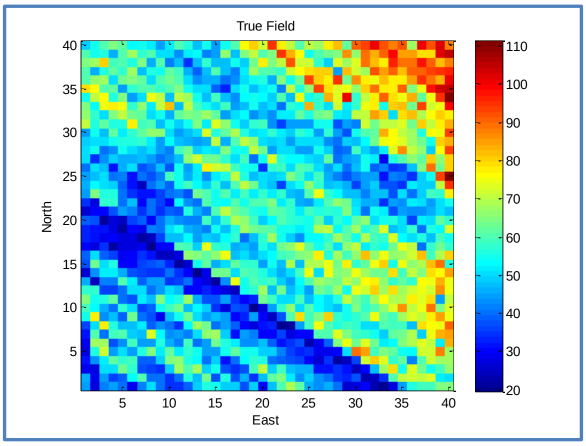
$$ax - range\ along\ the\ 135\ azimuth$$

$$ay - range\ along\ the\ 45\ azimuth$$

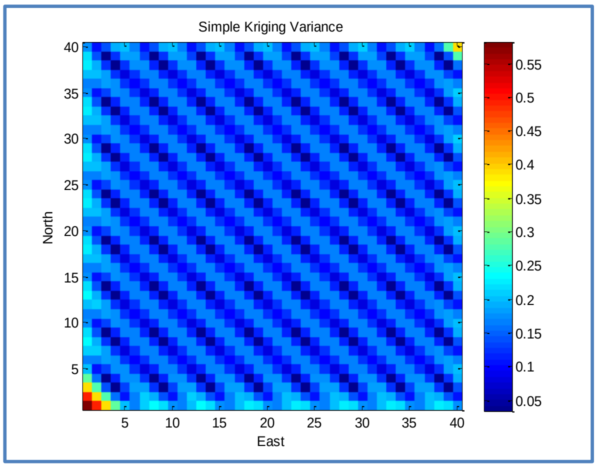
Q.3 The log(permeability) was normal score transformed and it was used for both simple and ordinary kriging. After kriging, the data was back transformed to the original values.

Simple Kriging



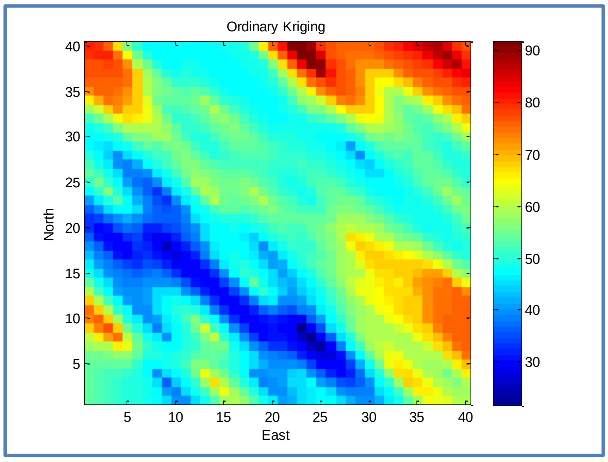


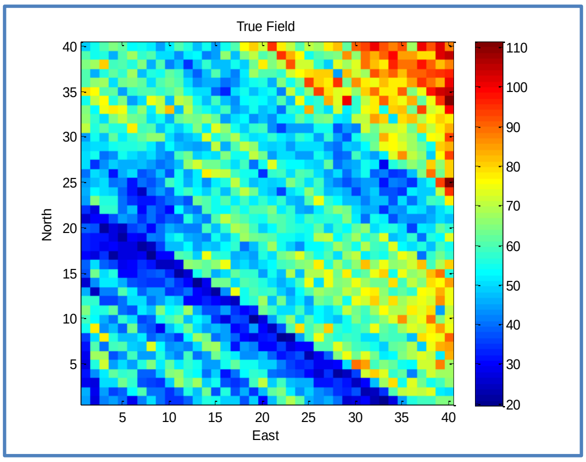
The simple kriging mean shows the values of permeability. The comparison b/w simple kriging and true field shows that kriging has captured the trends and the highs and lows of permeability very well. But the kriging estimate is very smooth as compared to the true field.



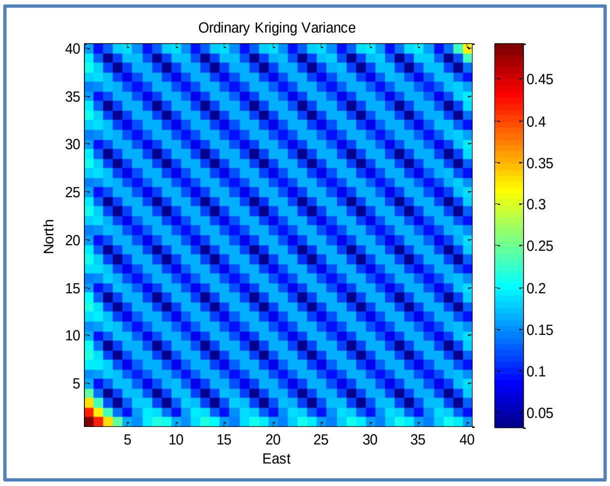
The simple kriging variance is in the normal score transformed space. The variance reduces to zero at the sample data locations.

Ordinary Kriging





The ordinary kriging mean shows the values of permeability. The ordinary kriging values are almost identical to the simple kriging values. The oridnary kriging also captures the trends present in the original data but is very smooth as compared to the original data.



The ordinary kriging variance is in the normal score transformed space. The variance reduces to zero at the sample data locations.

Part d)

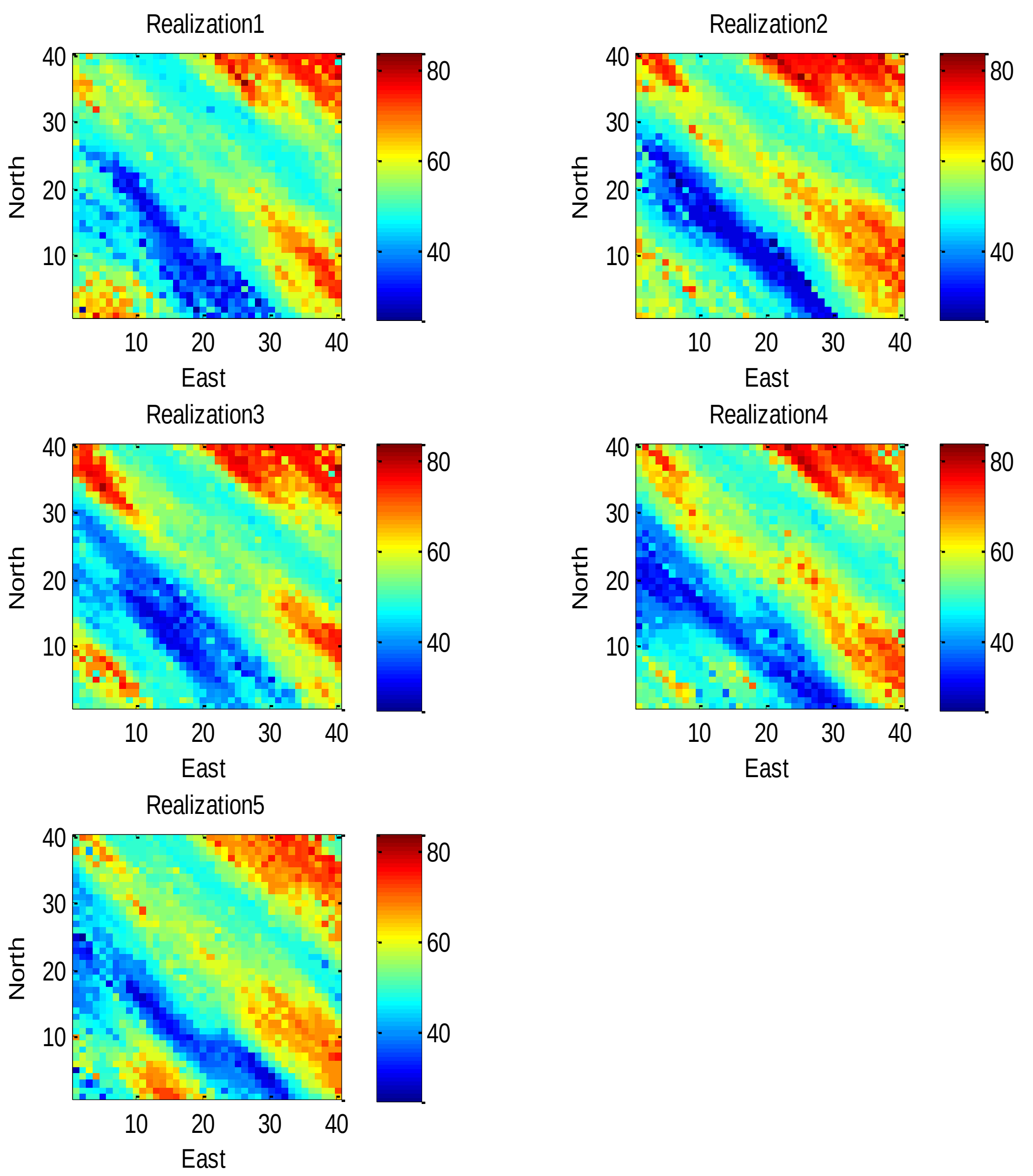
Comparison of Simple and Ordinary Kriging conclusions:

1) Both types of kriging yield almost the same results
2) The reason for this is that the original data is very dense.
3) The effect of normal score transforming the data for simple kriging removed the bias that a global mean estimate might have generated. This removal of bias also contributes to the almost identical results from simple and ordinary kriging.

Part e)

Sequential Gaussian Simulation (SGS)

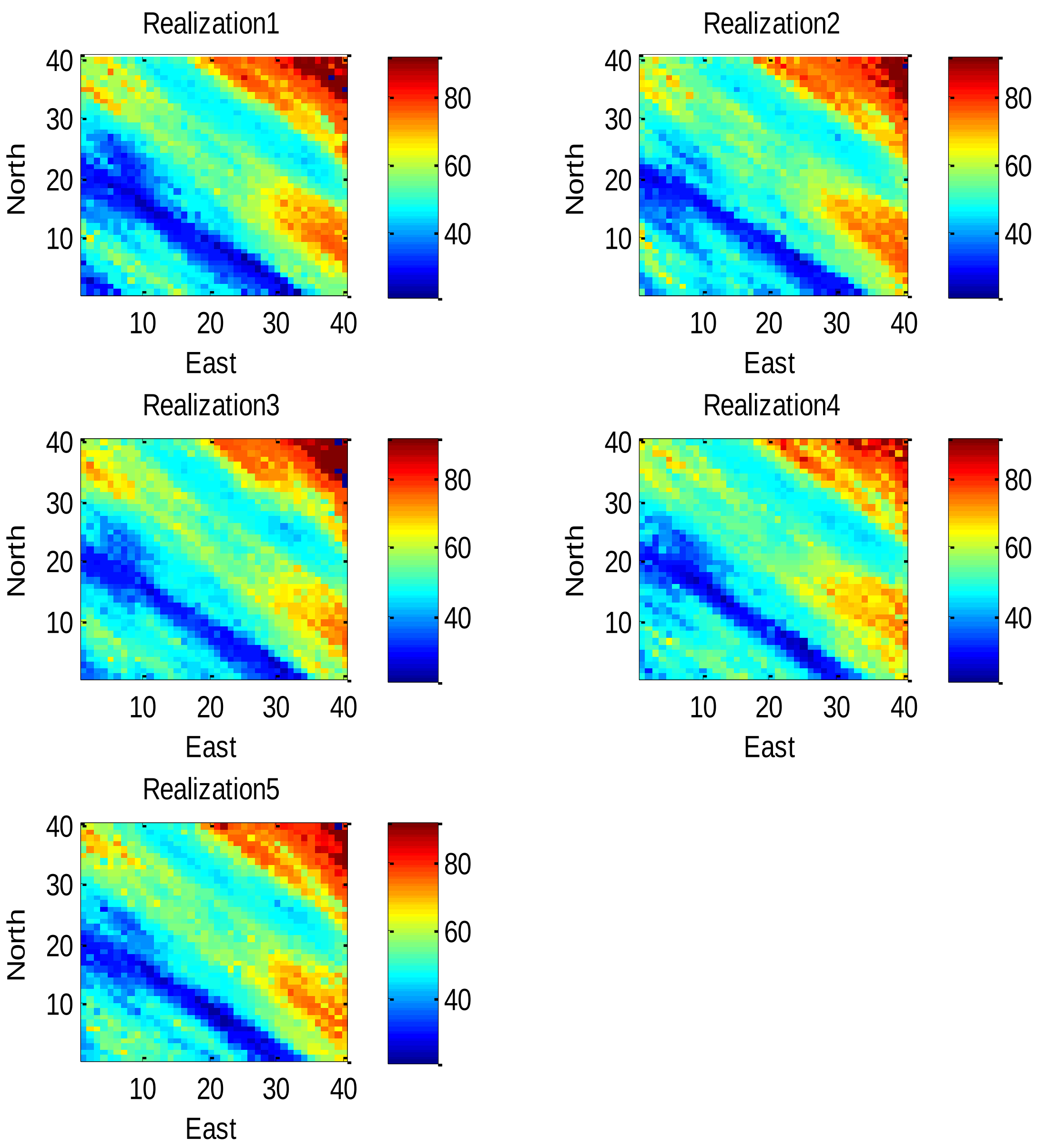
The log(permeability) was normal score transformed and it was used for generating the simulations. The data already computed was used for condition for the next unknown point. But, a limitation of 40 nearest points was placed for computation of the kriging matrix. The 4 nearest points were neglected in the computation of the kriging matrix because extremely near data was screening out the influence of other data in the search ellipsoid.



The different realizations vary quite a lot amongst themselves. **These realizations are not very smooth as compared to the previously generated kriging results.**

Part f)

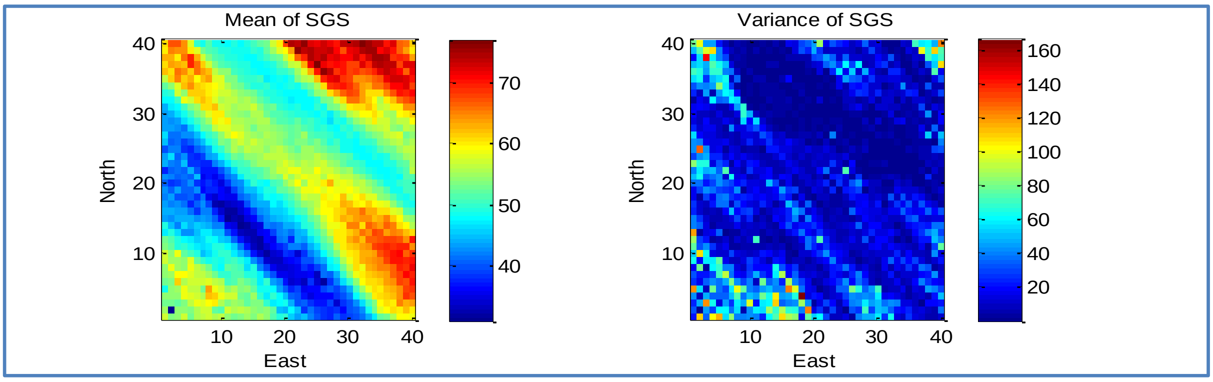
SGS with Bayesian Updating (SGSCOSIM)

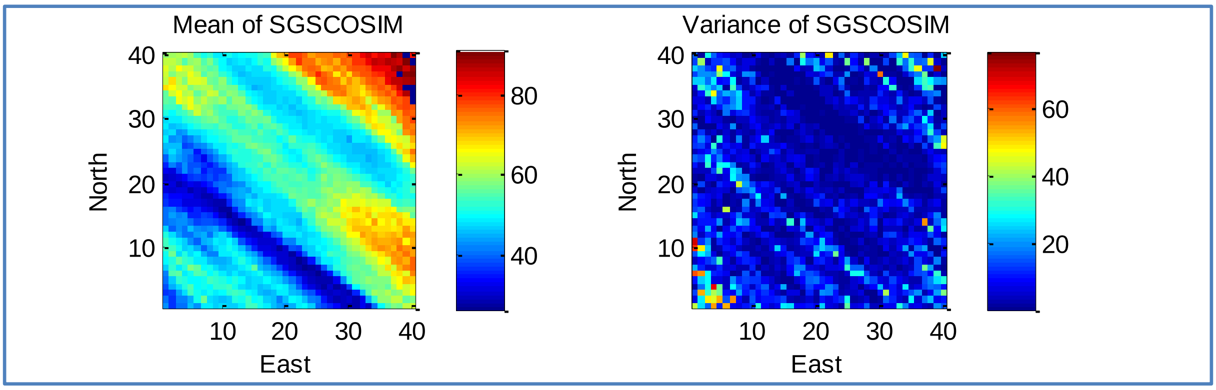


The realizations are not smooth as compared to kriging.

One difference with the previous realizations is that the **variation among the realizations is considerably reduced.**

Comparison of SGS and SGSCOSIM





1) The variance of SGS co-simulation is considerably smaller than that of SGS even with only 5 realizations.

2) The mean of SGSCOSIM is much nearer to the true data as compared to SGS.

CODE for SGSCOSIM

SGS with Bayesian updating

```
NST;%%normal score transform the data
nrlzn=5;%%% number of realizations

for t=1:nrlzn
```

```
Data=Data_nst;
Nx=40;
Ny=40;
dx=1;
dy=1;
L=[1:Nx*Ny];
Data_SGS=Data;%%initialize the data structure to which
%new conditioning data will be added
Data_SGS.varperm=zeros(length(Data.x),1);
sill=1;
rho=0.8; %coefficient of correlation

iter=length(Data.x);
```

start SGS loop

```
while L %%while there are locations left in grid
```

```
%%choose 1st location
    pos=randi(length(L));
    index=L(pos);
    L(pos)=[];
```

calculate x and y index and co-ordinates of randomly selected grid centre

```
if mod(index,Nx)==0
    xindex=Nx;
else
    xindex=mod(index,Nx);
end

yindex=((index-xindex)/Ny) + 1;

x_coord=(xindex-1)*dx+dx/2;
y_coord=(yindex-1)*dy+dy/2;
```

extract the k nearest neighbours

```
N=40;
List=[Data_SGS.x Data_SGS.y];
Query=[x_coord y_coord];
idx=knnsearch(List,Query,'k',N);
idx=idx(5:end);
N=N-4;
```

extract a matrix of the known nearest points which will be used for kriging

```
Dummy.x=Data_SGS.x(idx);
Dummy.y=Data_SGS.y(idx);
Dummy.lnperm=Data_SGS.lnperm(idx);
Dummy.varperm=Data_SGS.varperm(idx);
```

formulate the left hand side matrix for kriging

```
for i=1:N
    for j=1:N
        Coord1=[Dummy.x(i) Dummy.y(i)];
        Coord2=[Dummy.x(j) Dummy.y(j)];

        % get covariance from function
        cov=vargm(Coord1,Coord2);
        A(i,j)=cov;
    end
end
```

cholesky decomposition of A

```
LA=chol(A,'lower');
```

calculate grid centers

```
grid_dimensions
```

formulate the right hand matrix for all the grid centers

```
for j=1:N
      Coord1=[x_coord y_coord];
      Coord2=[Dummy.x(j) Dummy.y(j)];

      %get covariance
      cov=vargm(Coord1,Coord2);

      B(j,1)=cov;
end

  %solve Lz=B;
 z=LA\B;

 %solve L'x=z;
 lambda=LA'\z;

 lambda0=u*(1-sum(lambda));
 SK_est=lambda0+ lambda'*Dummy.lnperm;%%kriging mean
 SK_var=sill- lambda'*B;%% kriging variance
```

bayesian update

```
 zi=Data_sec_nst(yindex,xindex);
 SK_est=(rho*zi*SK_var+ SK_est*(1-rho^2))/(rho^2*(SK_var-1)+1);%%updated mean
 SK_var=SK_var*(1-rho^2)/(rho^2*(SK_var-1)+1);%%updated variance

 %%Random selction of location data value
 p=rand(1); %% choose a random value b/w 0&1;
 Gauss_est=norminv(p,SK_est,SK_var);
```

update dataset to include newly calculated data value

```
 iter=iter+1;
 Data_SGS.x(iter,1)=x_coord;
 Data_SGS.y(iter,1)=y_coord;
 Data_SGS.xindex(iter,1)=xindex;
 Data_SGS.yindex(iter,1)=yindex;
 Data_SGS.lnperm(iter,1)=Gauss_est;
 Data_SGS.varperm(iter,1)=SK_var;

 iter-64
```

```
end

RLZN=zeros(Ny,Nx);
  for i=65:iter
     %assign the dat values to the location on the grid for a particular
     %realization
     xindex=Data_SGS.xindex(i);
     yindex=Data_SGS.yindex(i);
     lnperm=Data_SGS.lnperm(i);

     %back transforming to original data space
     pt=normcdf(lnperm);
     lnperm=interp1(P,rearrdata,pt);
```

```
        RLZN(yindex,xindex)=lnperm;
    end
    Realization(t).RLZN=exp(RLZN);%% convert to permeability
    disp('realization complete');

end
```

plotting

```
for i=1:nrlzn
     subplot(3,2,i);
     imagesc((Realization(i).RLZN));
     if i==1
        cl = caxis; %# get color limits from the 1st image
    else
        caxis(cl) %# apply the same color limits to other images
    end
set(gca,'YDir','Normal');
xlabel('East');
ylabel('North');
s=strcat('Realization',num2str(i));
title(s);
colorbar;
 end
```

Published with MATLAB® R2013a