Contents

Optimal Interpolation Homework

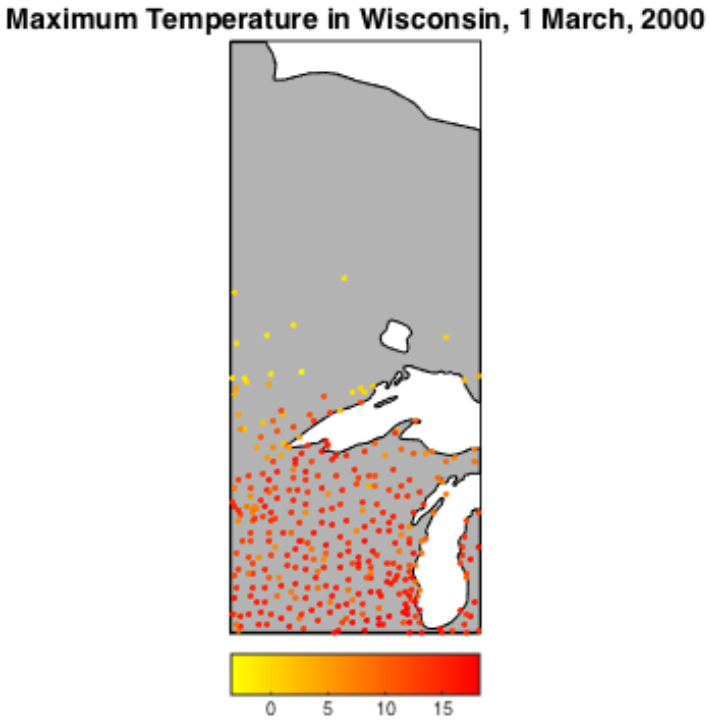
```
%%Scott Farley
%%May 2016
%
close all;
clear;
```

Load the data for Wisconsin

Plot the data using the station data points

```
latlim = [41.5 58.1] ;
lonlim = [-93.9 -85.8] ;

figure

colormap autumn;
colormap(flipud(colormap));

load '/Users/scottsfarley/downloads/tmax_1mar2000.mat'
ind = find( (lat >= latlim(1)) & (lat <= latlim(2)) & (lon >= lonlim(1)) & (lon <= lonlim(2)) ) ;
lat = double(lat(ind)) ;
lon = double(lon(ind)) ;
tmax = double(tmax(ind)) ;
tmax = tmax';
lat = lat';
lon = lon';
npt = length(ind) ;

land = shaperead('landareas.shp', 'UseGeoCoords', true) ;

axesm('mapprojection', 'eqdcylin', 'origin', [0 mean(lonlim)], ...
    'maplatlimit', latlim, 'maplonlimit', lonlim) ;
geoshow(land, 'FaceColor', 0.7*[1 1 1], 'EdgeColor', 'k') ;
scatterm(lat, lon, 10, tmax, 'filled') ;
caxis([min(tmax) max(tmax)]) ;
tightmap
colorbar('SouthOutside') ;
```

```
title('Maximum Temperature in Wisconsin, 1 March, 2000', 'fontsize', 14) ;

%measurement error
noiseamp = 0.4 ;
tmax = tmax + noiseamp*randn(npt, 1) ;
```

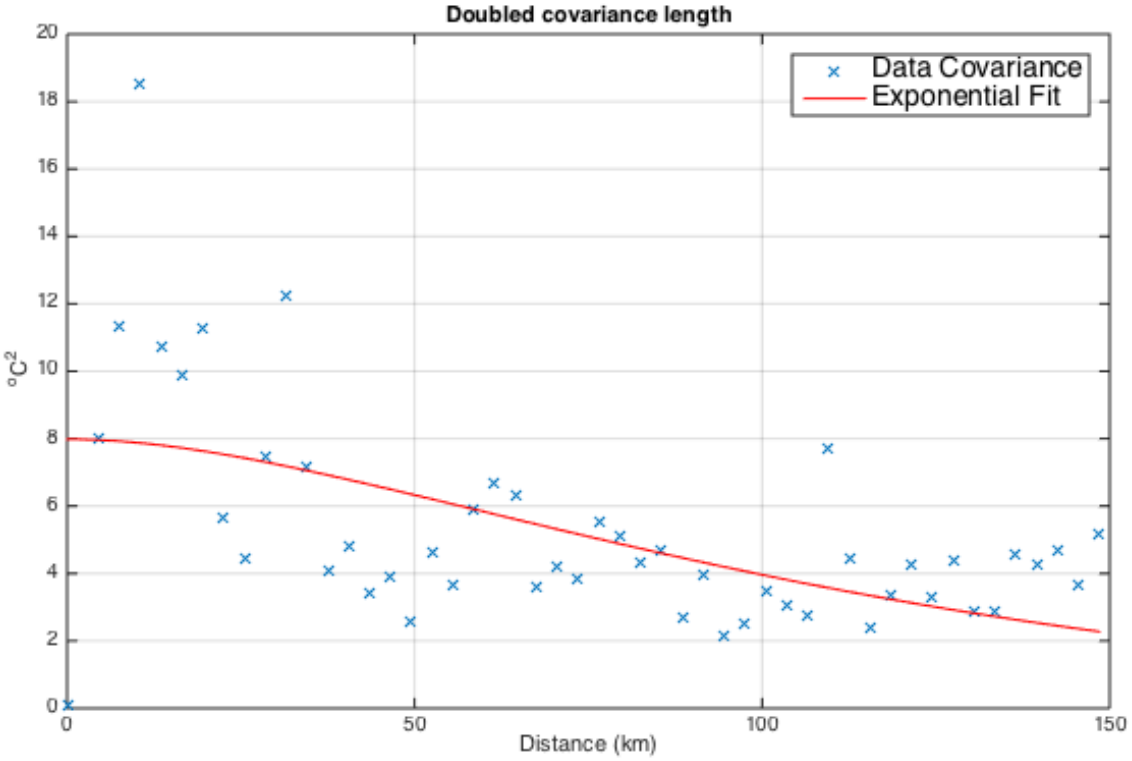


Parameterize the spatial covariance in the data

```
[xx, yy] = meshgrid(lon, lat);

xx = diag(xx);
yy = diag(yy);

rearth = 6370800 ;
xkm = (rearth * pi / 180) * (xx - mean(lonlim)) .* cos(yy*pi/180) / 1000 ;
ykm = (rearth * pi / 180) * (yy - mean(latlim)) / 1000 ;

Rdd = sqrt( ( (xkm * ones(1, npt)) - (ones(npt, 1) * xkm') ).^2 + ...
            ( (ykm * ones(1, npt)) - (ones(npt, 1) * ykm') ).^2 ) ;
Rdd = triu(Rdd) ; % Only keep the upper left triangle, as the matrix is symmetric

dr = 3; % bin size
bins = [0 (dr/2):dr:150] ;
C = (tmax - mean(tmax)) * (tmax - mean(tmax))' ;
cf = zeros(size(bins)) ;
cf(1) = mean(diag(C)) ;
for i = 1:length(bins) ;
    ind = find( (Rdd > bins(i)-dr/2) & (Rdd < bins(i)+dr/2) ) ;
    cf(i) = mean(C(ind)) ;
end
%
figure
```

```
pp = plot(bins, cf, 'x') ;
grid on ;
xlabel('Distance (km)') ;
ylabel('^oC^2') ;
% title('Binned lagged covariance with bin size = 25: Exponential') ;

ind = find(cf > 0) ;

% Gaussian function:
% myfun = @(a,x) a(1)^2 * exp(-x.^2/(a(2)^2)) ;
% "Markov" function - red noise:
myfun = @(a,x) a(1)*(1+x/a(2)).*exp(-x/a(2)) ;
% Exponential function
% myfun = @(a,x) a(1) * exp(-x / a(2)) ;

myerr = @(a,x,y) sum( (myfun(a,x) - y).^2 ) ;
a = [1 600]

myerr(a,bins(ind),cf(ind)) / length(bins)

b = fminsearch(@(a) myerr(a,bins(ind),cf(ind)), [1 500]) ;
title('Doubled covariance length');



hold on ;
pp2 = plot(bins, myfun(b,bins), 'r') ;
hold off ;

ll = legend([pp pp2], 'Data Covariance', 'Exponential Fit') ;
set(ll, 'fontsize', 14) ;
```

```
a =

     1   600


ans =

   27.5000
```



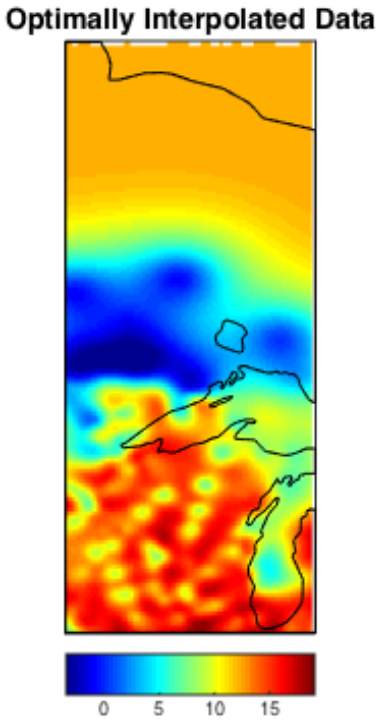

```
Rdd = sqrt( ( (xkm * ones(1, npt)) - (ones(npt, 1) * xkm') ).^2 + ...
            ( (ykm * ones(1, npt)) - (ones(npt, 1) * ykm') ).^2 ) ;

Cdd0 = myfun(b,Rdd)/b(1) ;


dx = 0.1 ;
dy = 0.1 ;
[xgrd, ygrd] = meshgrid( ...
    [lonlim(1):dx:lonlim(2)]', [latlim(1):dy:latlim(2)]) ;
xgkm = (rearth * pi / 180) * (xgrd(:) - mean(lonlim)) .* cos(ygrd(:)*pi/180) / 1000 ;
ygkm = (rearth * pi / 180) * (ygrd(:) - mean(latlim)) / 1000 ;
nptg = length(xgkm) ;

Rgd = sqrt( ( (xgkm * ones(1, npt)) - (ones(nptg, 1) * xkm') ).^2 + ...
            ( (ygkm * ones(1, npt)) - (ones(nptg, 1) * ykm') ).^2 ) ;
Cgd0 = myfun(b, Rgd)/b(1) ;

%%Do the optimal Interpolation
tic
 D = mean(tmax) + Cgd0*inv(Cdd0 + noiseamp^2/b(1)*eye(npt))*(tmax-mean(tmax));
toc

D = reshape(D, size(xgrd)) ;
```

```
Elapsed time is 0.156617 seconds.
```

plot the optimal Interpolation

```
figure
```

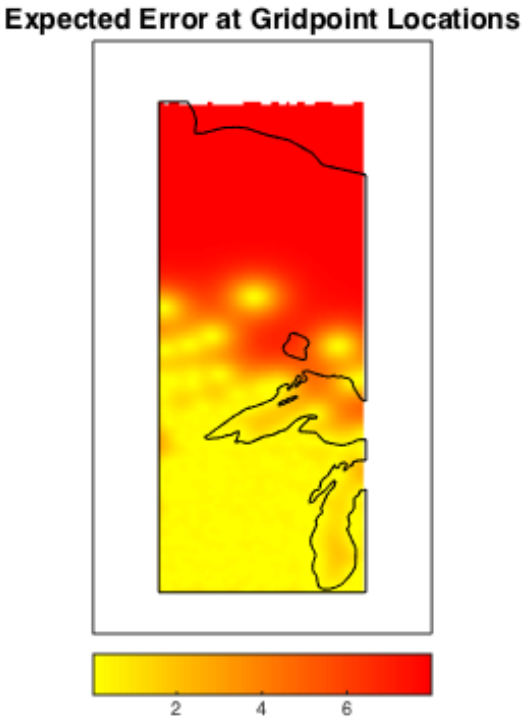
```
colormap jet;
%colormap(flipud(colormap));
axesm('mapprojection', 'eqdcylin', 'origin', [0 mean(lonlim)], ...
    'maplatlimit', latlim, 'maplonlimit', lonlim) ;
surfacem(ygrd, xgrd, D) ;
hold on ;
hold off ;
caxis([min(tmax) max(tmax)]) ;
geoshow(land, 'FaceColor', 'none', 'EdgeColor', 'k') ;
colorbar('SouthOutside') ;
tightmap
title('Optimally Interpolated Data', 'fontsize', 14) ;
```



Plot expected errors

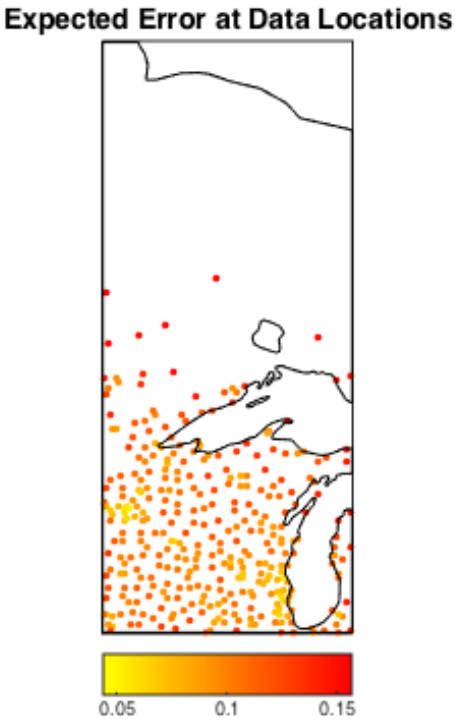
```
oierror = b(1)*diag(1-Cgd0*inv(Cdd0 + noiseamp^2/b(1)*eye(npt))*Cgd0');
error0 = b(1)*diag(1-Cdd0*inv(Cdd0 + noiseamp^2/b(1)*eye(npt))*Cdd0');
```

```
figure
colormap autumn;
colormap(flipud(colormap));
axesm('mapprojection', 'eqdcylin', 'origin', [0 mean(lonlim)], ...
    'maplatlimit', latlim, 'maplonlimit', lonlim) ;
surfacem(ygrd, xgrd, reshape(oierror, size(xgrd))) ;
colorbar('SouthOutside') ;
geoshow(land, 'FaceColor', 'none', 'EdgeColor', 'k') ;
title('Expected Error at Gridpoint Locations', 'fontsize', 14) ;
```




```
figure
colormap autumn;
colormap(flipud(colormap));
axesm('mapprojection', 'eqdcylin', 'origin', [0 mean(lonlim)], ...
    'maplatlimit', latlim, 'maplonlimit', lonlim) ;
hh = scatterm(yy, xx, 15, error0, 'filled');
caxis([min(error0) max(error0)]) ;
tightmap
colorbar('SouthOutside') ;
geoshow(land, 'FaceColor', 'none', 'EdgeColor', 'k') ;
title('Expected Error at Data Locations', 'fontsize', 14) ;

mean(error0)
mean(oierror)
```

```
ans =

    0.1032


ans =

    3.9733
```



Washington!

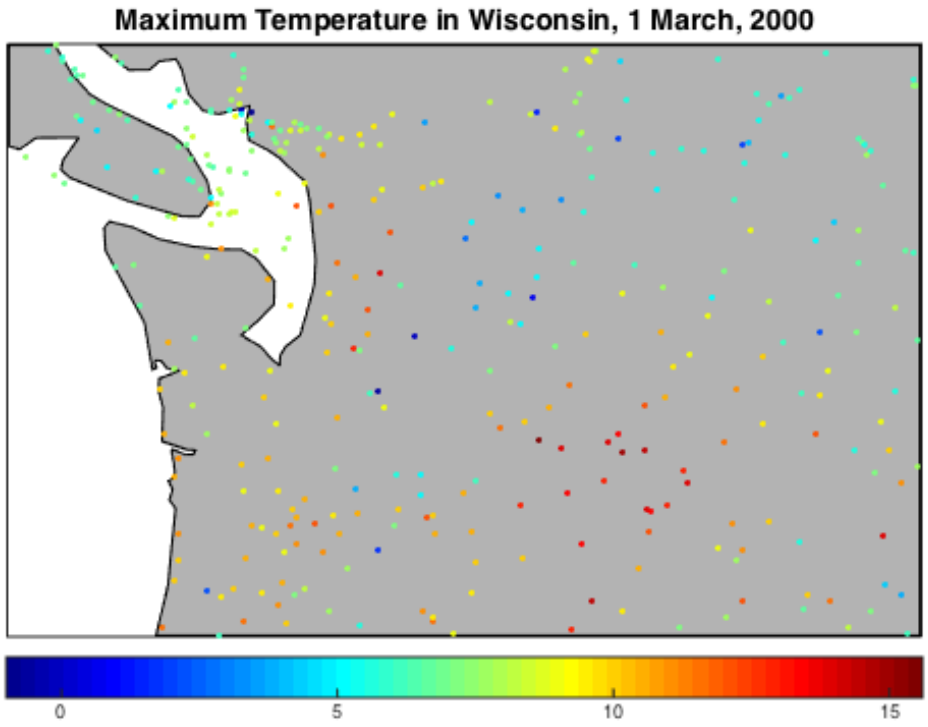
```
close all;
clear;
```

Load the data for Washington

Plot the data using the station data points

```
latlim = [44.5  50.0] ;
lonlim = [-125.7 -115.9] ;


figure

colormap jet;
% colormap(flipud(colormap));

load '/Users/scottsfarley/downloads/tmax_1mar2000.mat'
ind = find( (lat >= latlim(1)) & (lat <= latlim(2)) & (lon >= lonlim(1)) & (lon <= lonlim(2)) ) ;
lat = double(lat(ind)) ;
lon = double(lon(ind)) ;
tmax = double(tmax(ind)) ;
npt = length(ind) ;

land = shaperead('landareas.shp', 'UseGeoCoords', true) ;

axesm('mapprojection', 'eqdcylin', 'origin', [0 mean(lonlim)], ...
    'maplatlimit', latlim, 'maplonlimit', lonlim) ;
geoshow(land, 'FaceColor', 0.7*[1 1 1], 'EdgeColor', 'k') ;
scatterm(lat, lon, 10, tmax, 'filled') ;
```

```
caxis([min(tmax) max(tmax)]) ;
tightmap
colorbar('SouthOutside') ;
title('Maximum Temperature in Wisconsin, 1 March, 2000', 'fontsize', 14) ;

%measurement error
tmax = tmax';
lat = lat';
lon = lon';
noiseamp = 0.4 ;
tmax = tmax + noiseamp*randn(npt, 1) ;
```

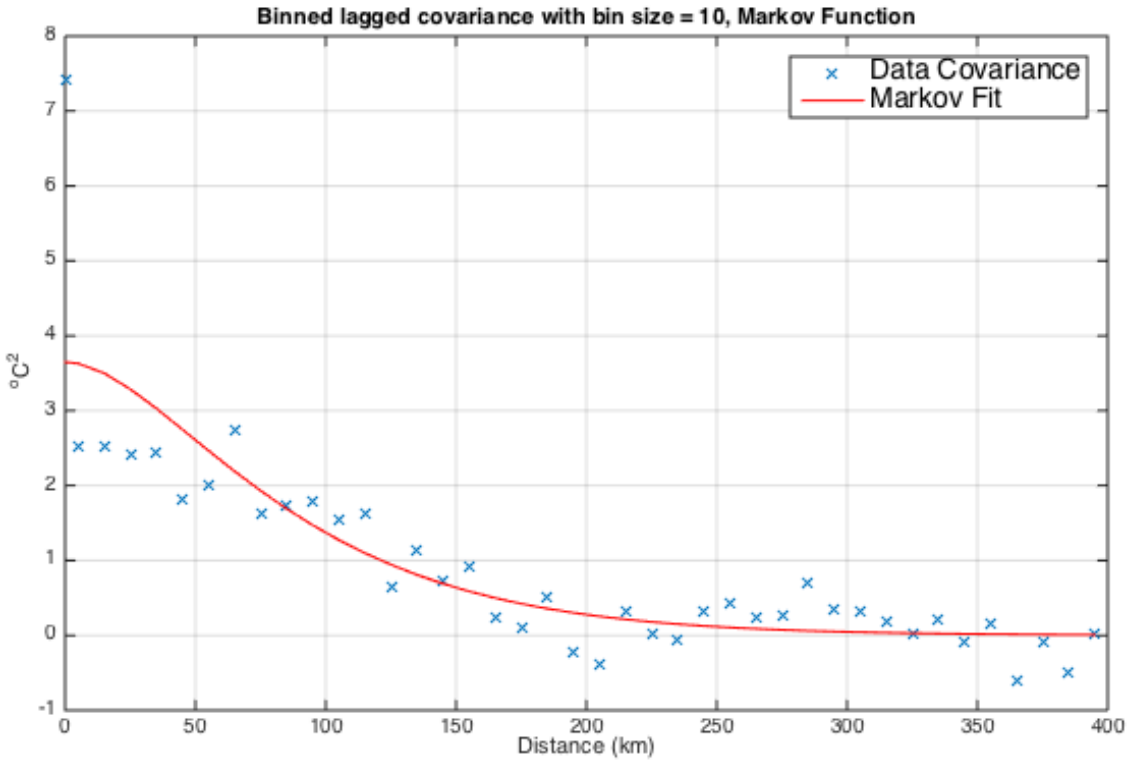


Parameterize the spatial covariance in the data

```
[xx, yy] = meshgrid(lon, lat);

xx = diag(xx);
yy = diag(yy);

rearth = 6370800 ;
xkm = (rearth * pi / 180) * (xx - mean(lonlim)) .* cos(yy*pi/180) / 1000 ;
ykm = (rearth * pi / 180) * (yy - mean(latlim)) / 1000 ;

Rdd = sqrt( ( (xkm * ones(1, npt)) - (ones(npt, 1) * xkm') ).^2 + ...
            ( (ykm * ones(1, npt)) - (ones(npt, 1) * ykm') ).^2 ) ;
Rdd = triu(Rdd) ; % Only keep the upper left triangle, as the matrix is symmetric

dr = 10; % bin size
bins = [0 (dr/2):dr:400] ;
C = (tmax - mean(tmax)) * (tmax - mean(tmax))' ;
cf = zeros(size(bins)) ;
cf(1) = mean(diag(C)) ;
```

```
for i = 2:length(bins) ;
    ind = find( (Rdd > bins(i)-dr/2) & (Rdd < bins(i)+dr/2) ) ;
    cf(i) = mean(C(ind)) ;
end
%
figure
pp = plot(bins, cf, 'x') ;
grid on ;
xlabel('Distance (km)') ;
ylabel('^oC^2') ;
title('Binned lagged covariance with bin size = 10, Markov Function') ;

ind = find(cf > 0) ;

% Gaussian function:
% myfun = @(a,x) a(1)^2 * exp(-x.^2/(a(2)^2)) ;
% "Markov" function - red noise:
myfun = @(a,x) a(1)*(1+x/a(2)).*exp(-x/a(2)) ;
% Exponential function
% myfun = @(a,x) a(1) * exp(-x / a(2)) ;

myerr = @(a,x,y) sum( (myfun(a,x) - y).^2 ) ;

options = optimset('MaxFunEvals', 1000000000000);
b = fminsearch(@(a) myerr(a,bins(ind),cf(ind)), [1 500], options) ;

myerr = @(a,x,y) sum( (myfun(a,x) - y).^2 ) ;
a = [1 600]

myerr(a,bins(ind),cf(ind)) / length(bins)

hold on ;
pp2 = plot(bins, myfun(b,bins), 'r') ;
hold off ;

ll = legend([pp pp2], 'Data Covariance', 'Markov Fit') ;
set(ll, 'fontsize', 14) ;
```

```
a =

     1   600


ans =

    1.5826
```




```
b(1) % --> Variance at zero distance...intercept
```

```
ans =

    3.6493
```

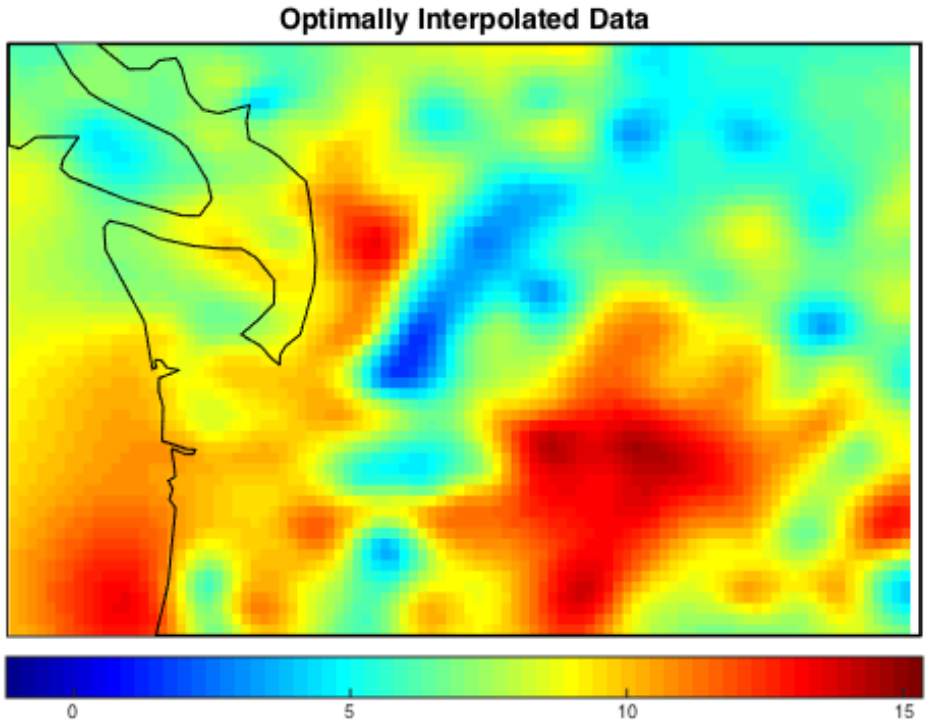
```
Rdd = sqrt( ( (xkm * ones(1, npt)) - (ones(npt, 1) * xkm') ).^2 + ...
            ( (ykm * ones(1, npt)) - (ones(npt, 1) * ykm') ).^2 ) ;

Cdd0 = myfun(b,Rdd)/b(1) ;

dx = 0.1 ;
dy = 0.1 ;
[xgrd, ygrd] = meshgrid( ...
    [lonlim(1):dx:lonlim(2)]', [latlim(1):dy:latlim(2)]) ;
xgkm = (rearth * pi / 180) * (xgrd(:) - mean(lonlim)) .* cos(ygrd(:)*pi/180) / 1000 ;
ygkm = (rearth * pi / 180) * (ygrd(:) - mean(latlim)) / 1000 ;
nptg = length(xgkm) ;

Rgd = sqrt( ( (xgkm * ones(1, npt)) - (ones(nptg, 1) * xkm') ).^2 + ...
            ( (ygkm * ones(1, npt)) - (ones(nptg, 1) * ykm') ).^2 ) ;
Cgd0 = myfun(b, Rgd)/b(1) ;

%%Do the optimal Interpolation
tic
 D = mean(tmax) + Cgd0*inv(Cdd0 + noiseamp^2/b(1)*eye(npt))*(tmax-mean(tmax));
toc
```

```
D = reshape(D, size(xgrd)) ;
```

```
Elapsed time is 0.091748 seconds.
```

plot the optimal Interpolation

```
figure
colormap jet;
%colormap(flipud(colormap));
axesm('mapprojection', 'eqdcylin', 'origin', [0 mean(lonlim)], ...
    'maplatlimit', latlim, 'maplonlimit', lonlim) ;
surfacem(ygrd, xgrd, D) ;
hold on ;
hold off ;
caxis([min(tmax) max(tmax)]) ;
geoshow(land, 'FaceColor', 'none', 'EdgeColor', 'k') ;
colorbar('SouthOutside') ;
tightmap
title('Optimally Interpolated Data', 'fontsize', 14) ;
```

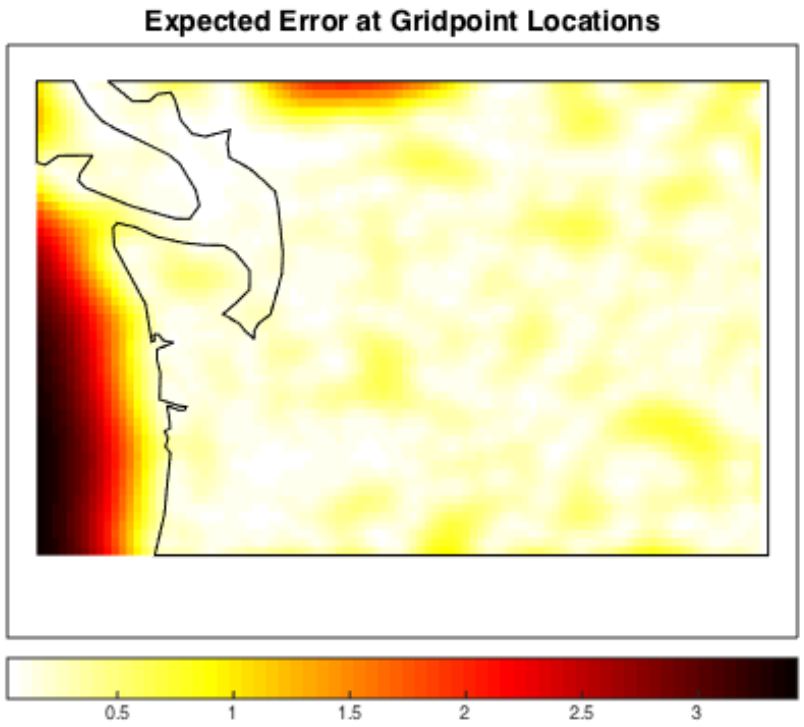


Plot expected errors

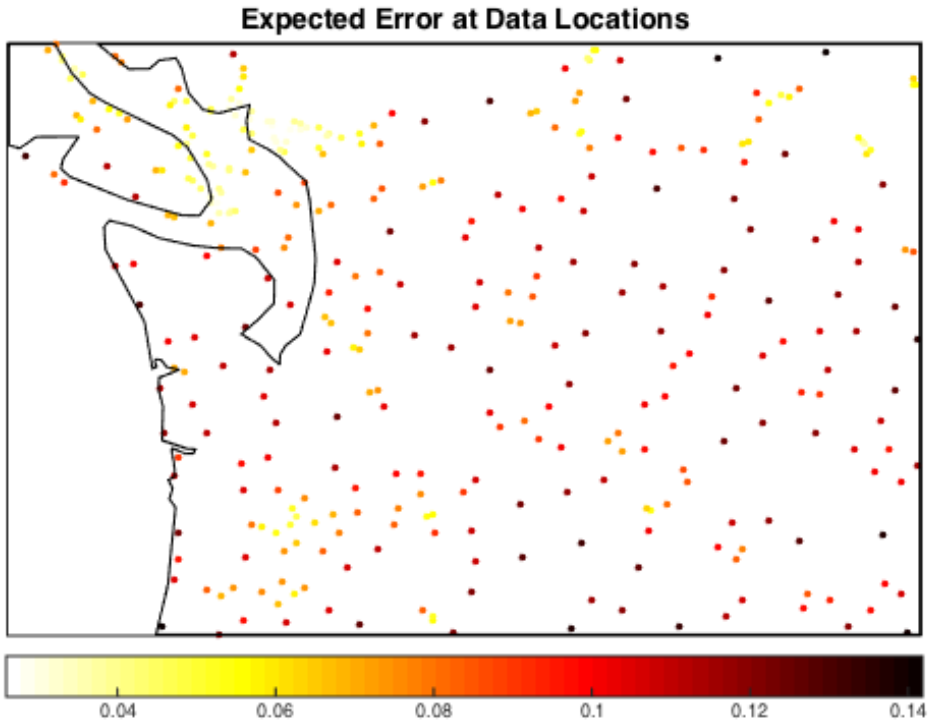
```
oierror = b(1)*diag(1-Cgd0*inv(Cdd0 + noiseamp^2/b(1)*eye(npt))*Cgd0');
error0 = b(1)*diag(1-Cdd0*inv(Cdd0 + noiseamp^2/b(1)*eye(npt))*Cdd0');
```

```
figure
colormap hot;
```

```
colormap(flipud(colormap));
axesm('mapprojection', 'eqdcylin', 'origin', [0 mean(lonlim)], ...
    'maplatlimit', latlim, 'maplonlimit', lonlim) ;
surfacem(ygrd, xgrd, reshape(oierror, size(xgrd))) ;
colorbar('SouthOutside') ;
geoshow(land, 'FaceColor', 'none', 'EdgeColor', 'k') ;
title('Expected Error at Gridpoint Locations', 'fontsize', 14) ;
```




```
figure
colormap hot;
colormap(flipud(colormap));
axesm('mapprojection', 'eqdcylin', 'origin', [0 mean(lonlim)], ...
    'maplatlimit', latlim, 'maplonlimit', lonlim) ;
hh = scatterm(yy, xx, 15, error0, 'filled');
caxis([min(error0) max(error0)]) ;
tightmap
colorbar('SouthOutside') ;
geoshow(land, 'FaceColor', 'none', 'EdgeColor', 'k') ;
title('Expected Error at Data Locations', 'fontsize', 14) ;
```



Published with MATLAB® R2014b