Estimating decorrelation length scales

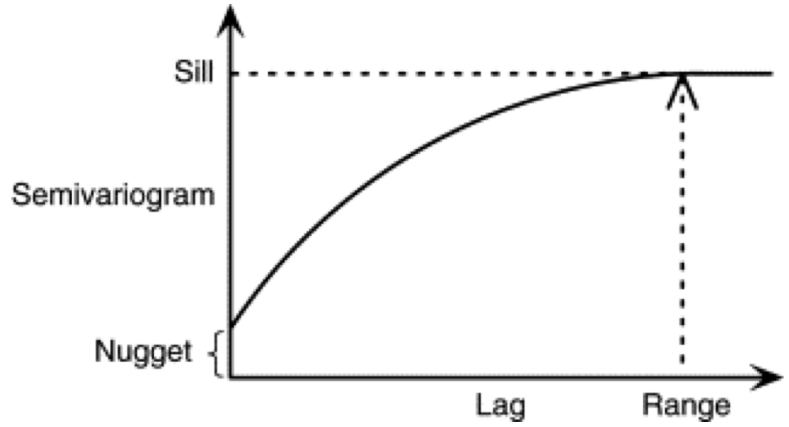
Semi-variogram

A semi-variogram plots semi-variance versus range to indicate the degree of spatial dependence between values of *Z* at two points in space separated by lag *h*.

$$\gamma(h) = \frac{1}{2N(h)} \sum_{i=1}^{N(h)} [Z(x_i) - Z(x_i + h)]^2$$

where *N(h)* is the number of data pairs separated by lag *h.*

At large lag the semi-variogram reaches a plateau often termed the *sill**. Beyond this range the semi-variance is just that due to uncorrelated variability.



A non-zero *nugget** can arise because repeated observations at zero lag have observation error.

By fitting a smooth function to an empirically estimated semi-variogram a *Range of Influence* parameter, *a*, gives an estimate of the scalar covariance length scale.

Popular functional fits to a semi-variogram are:

$$\gamma(h) = C_0 + C_1 \left\{1 - \exp\left(-\frac{h}{a}\right)\right\}$$

$$\gamma(h) = C_0 + C_1 \left\{1 - \exp\left(-\frac{h^2}{a^2}\right)\right\}$$

* This nomenclature is inherited from the geology community from techniques developed for mapping mineral resources.

Estimating decorrelation length scales

Covariance function

The semi-variogram is related to the auto-covariance of *Z* at points separated by a distance *h*.

$$\mathrm{Cov}\left[Z\left(\mathbf{s}_1\right), Z\left(\mathbf{s}_2\right)\right] = \mathrm{Cov}\left[Z\left(\mathbf{s}_1\right), Z\left(\mathbf{s}_1 + \mathbf{h}\right)\right] = \mathrm{C}\left(\mathbf{h}\right) = \mathrm{E}\left[Z\left(\mathbf{s}_1\right) Z\left(\mathbf{s}_1 + \mathbf{h}\right)\right]$$

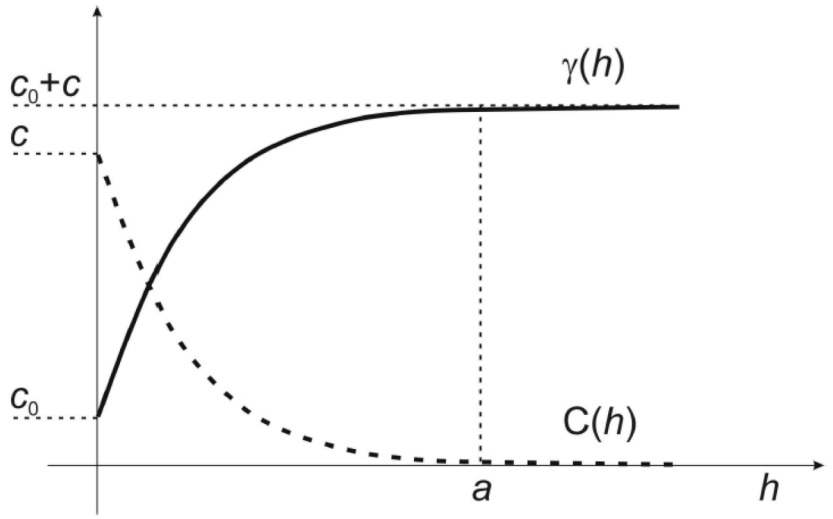


Fig. 1. The relation between the covariance function C(h) and semivariogram $\gamma(h)$ for second order stationary random fields

For some sample data set of observed or modeled values

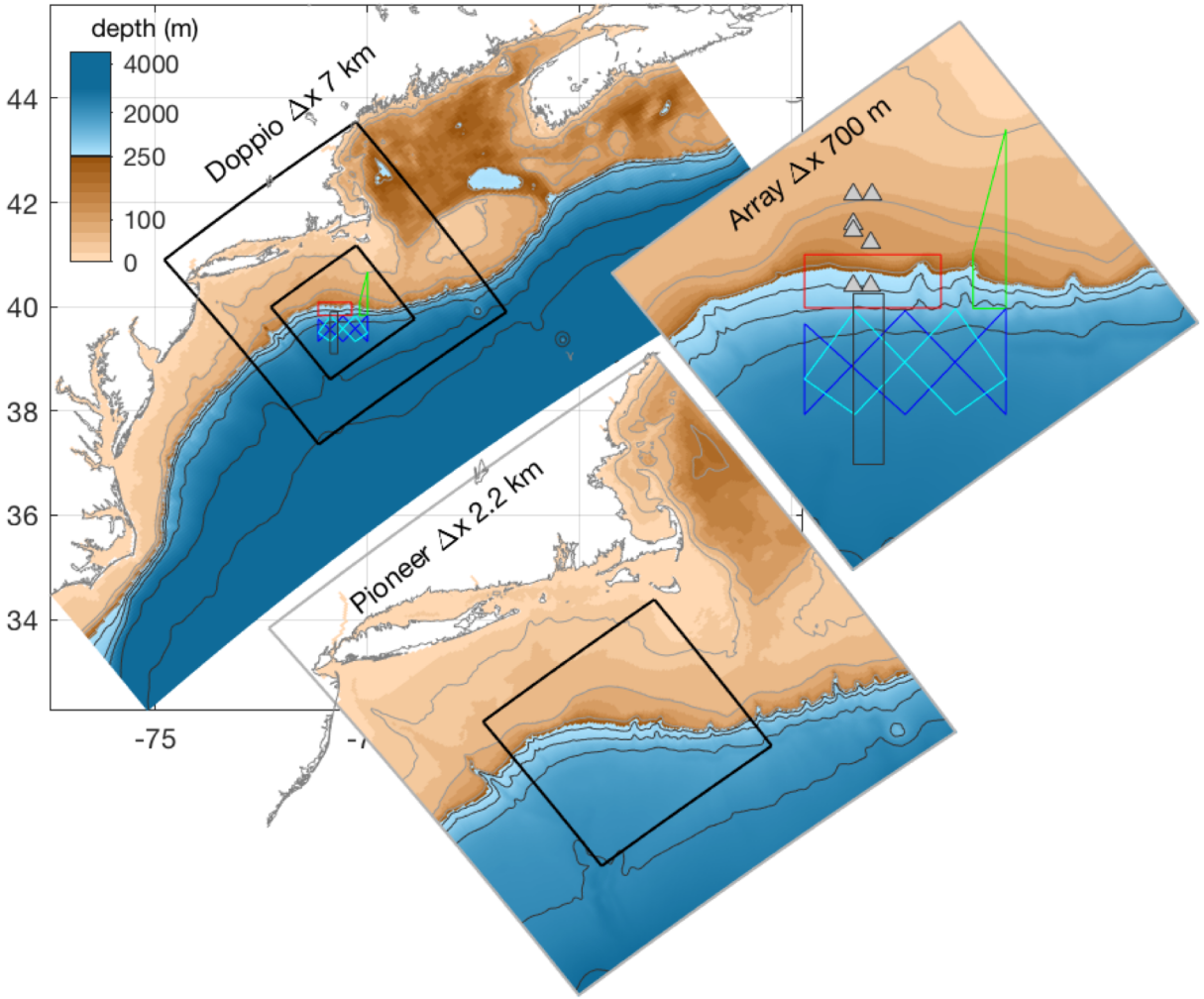
(i) Find the subset of pairs points separated by a certain distance (actually a small band centered on some nominal range)
(ii) Compute the semi-variance and/or covariance of the sample subset
(iii) Fit a function to the data and use the characteristic length scale to define the nominal length scale for the background error covariance

The particular approach to (i) will depend on whether you have arbitrarily distributed observations or gridded model or data.

Estimating decorrelation length scales

We compute semi-variograms and binned lagged covariance functions from the results of free running nested ROMS simulations.

- Array 700 m
- Pioneer 2.2 km
- Doppio 7 km



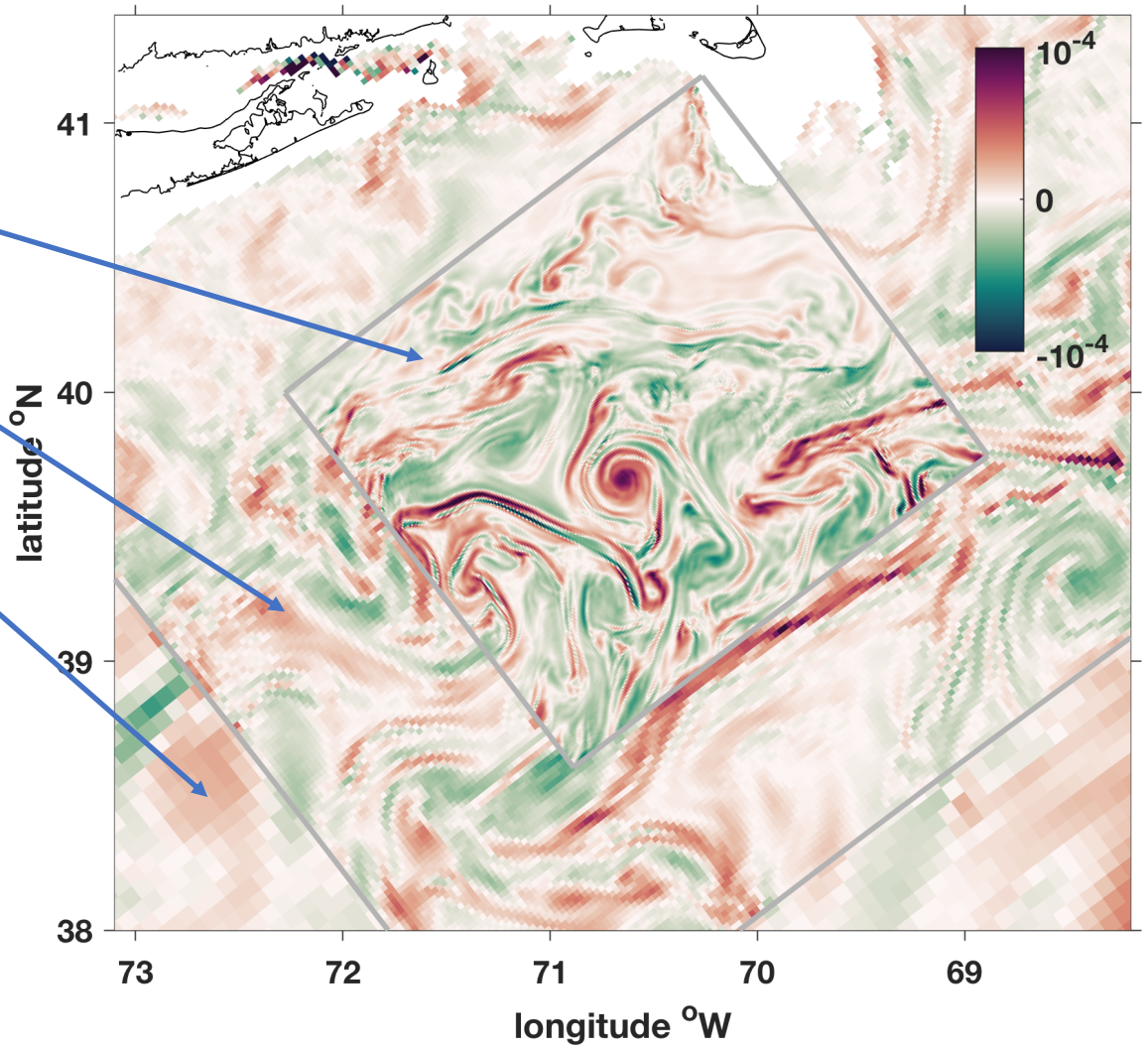
The model output is on a regular grid, so we can take a single snapshot and slide it *N* cells in the *i* direction to compute the semi-variance and covariance of the overlapping points to get an estimate at separation
*h = N*dx*

Estimating decorrelation length scales

We compute semi-variograms and binned lagged covariance functions from the results of free running nested ROMS simulations.

- Array 700 m
- Pioneer 2.2 km
- Doppio 7 km

The model output is on a regular grid, so we can take a single snapshot and slide it *N* cells in the *i* direction to compute the semi-variance and covariance of the overlapping points to get an estimate at separation
*h = N*dx*

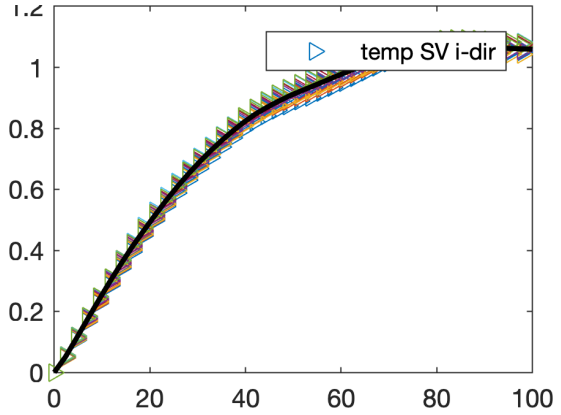


Estimating decorrelation length scales

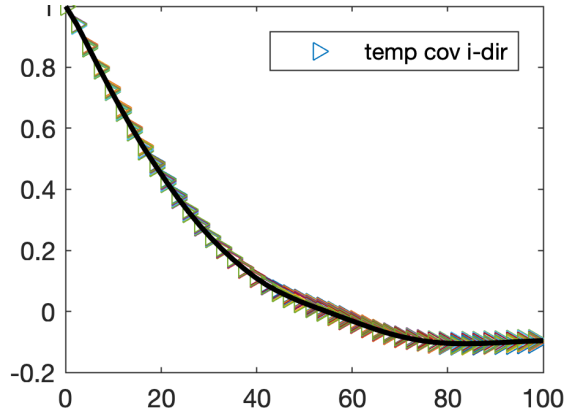
```
% shift scene by s pixels in i
d1 = data((s+1):end,:);
d2 = data(1:(end-s),:);

% semi-variogram
% one half mean squared difference
dd = d2-d1;
sv = 0.5*nanmean(dd(:).^2);

% covariance
dd = d2.*d1;
cv = nanmean(dd(:));

% assign value to bin
SVy(s+1) = sv;
CVy(s+1) = cv;
Rx(s+1) = s*DX;
```

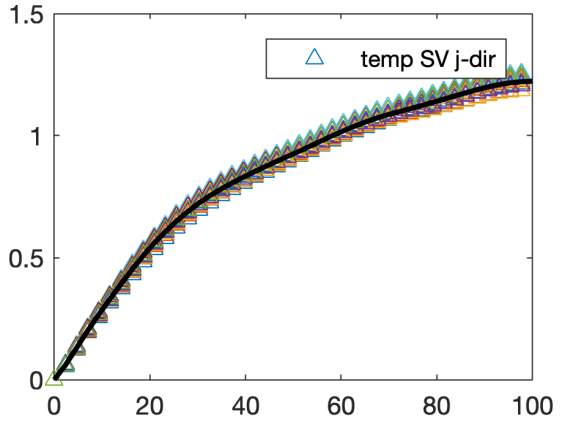
Results for “pioneer” model grid

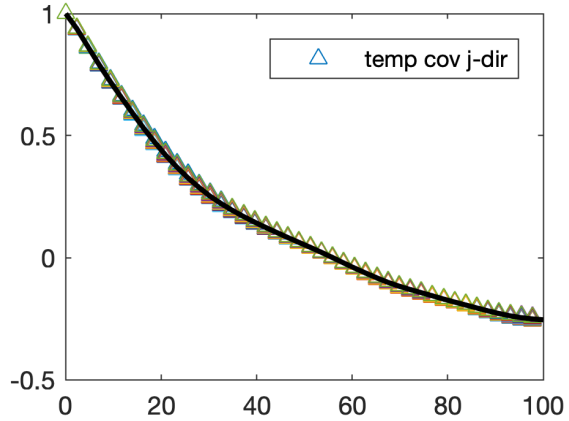
semi-variogram



covariance







Repeat for the *j-direction* and accumulate statistics over multiple time steps to obtain a smooth sample semi-variogram or covariance function …

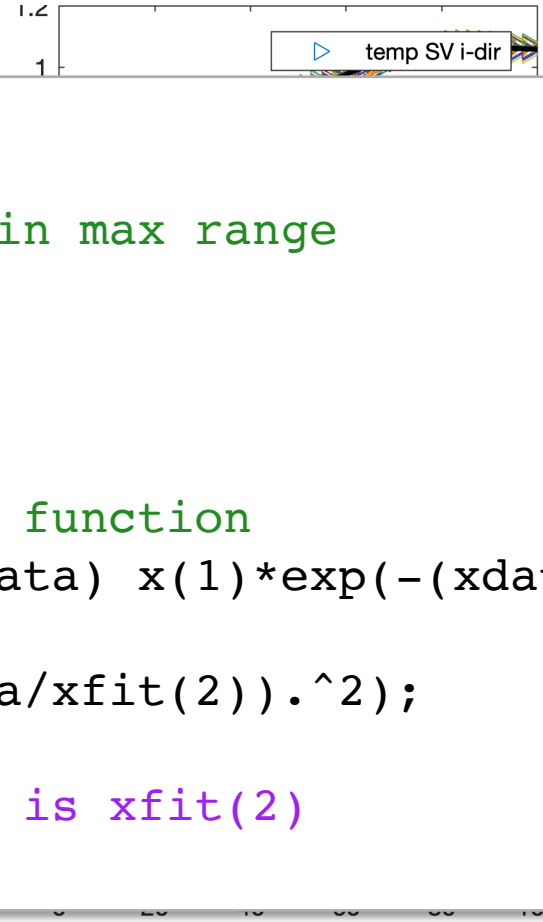
… Then fit a function

Estimating decorrelation length scales

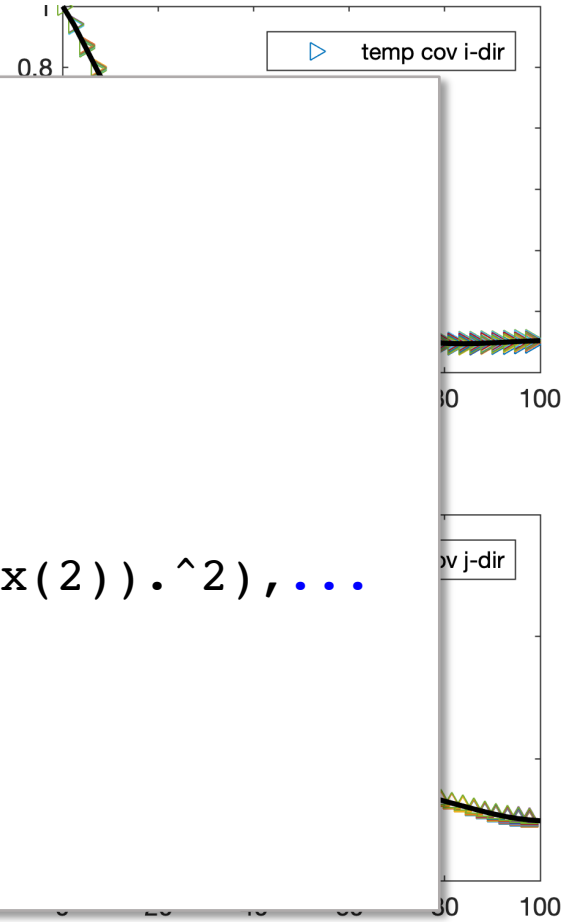
```
% shift scene by s pixels in i
d1 = data((s+1):end,:);
d2 = data(1:(end-s),:);

% semi-var
% one half
dd = d2-d1
sv = 0.5*n

% covarian
dd = d2.*d
cv = nanme

% assign v
SVy(s+1) =
CVy(s+1) =
Rx(s+1) =
```

Results for "pioneer" model grid

semi-variogram



covariance



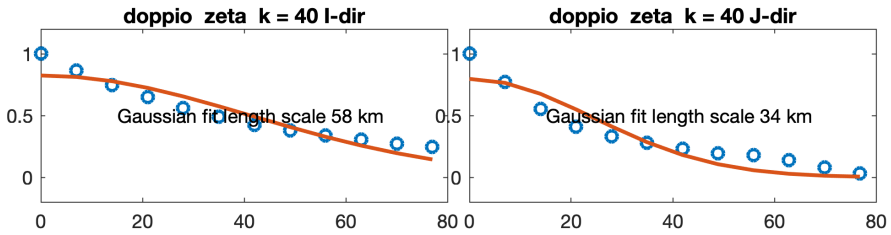

```
%% Use Matlab lsqcurvefit
% restrict fit to a certain max range
subset = find(Rx < Rmax);
xdata = Rx(subset);
ydata = CVx(subset);

% fit Gaussian covariance function
xfit = lsqcurvefit(@(x,xdata) x(1)*exp(-(xdata/x(2)).^2),...
  [1 50],xdata,ydata);
yfit = xfit(1)*exp(-(xdata/xfit(2)).^2);

Gaussian fit length scale is xfit(2)
```

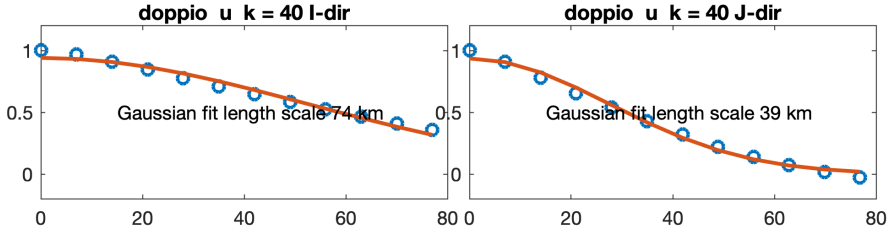
Repeat for the *j-direction* and accumulate statistics over multiple time steps to obtain a smooth sample semi-variogram or covariance function …

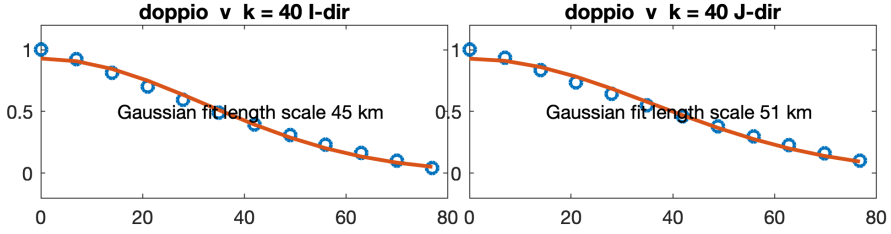
… Then fit a function

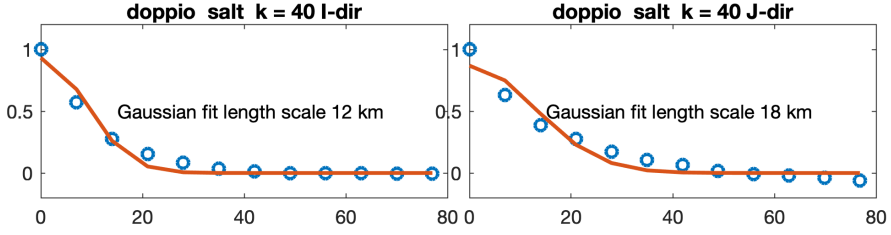
Fitted covariance functions for all variables

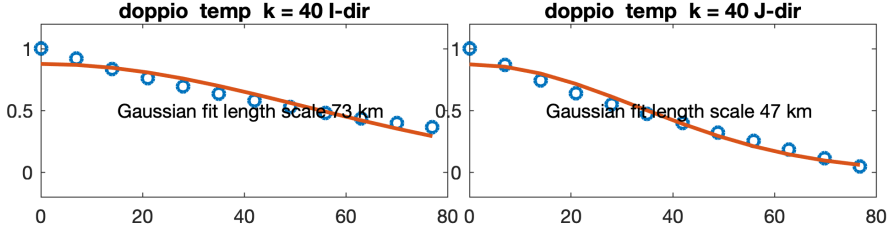
Doppio 7 km



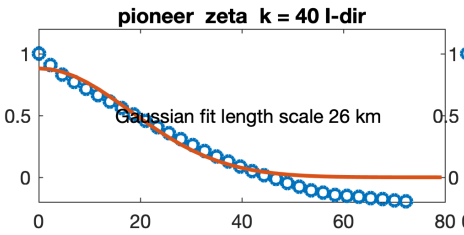


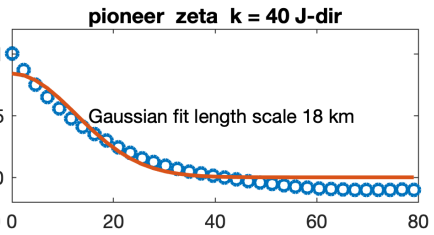


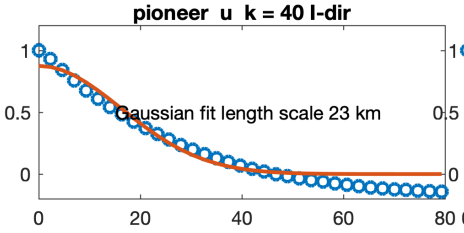


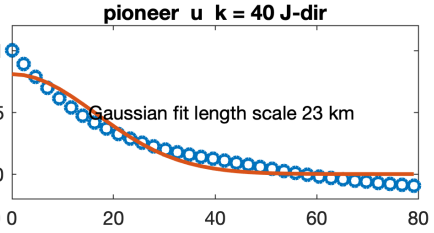


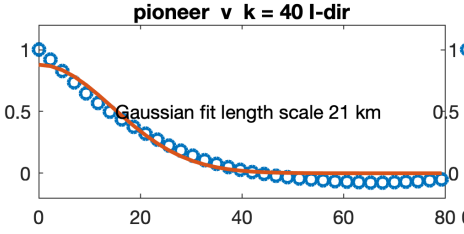
Pioneer 2.2 km

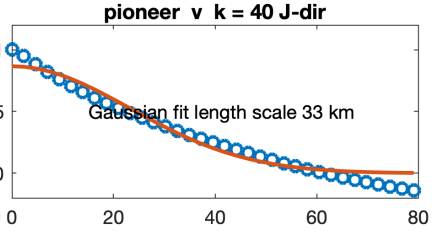


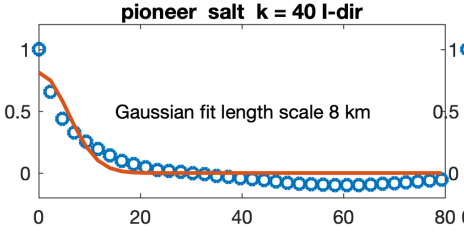


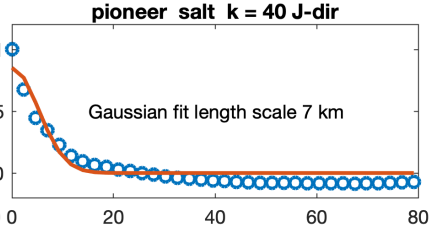


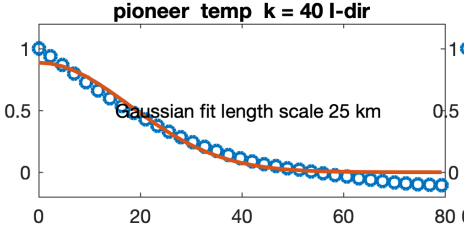


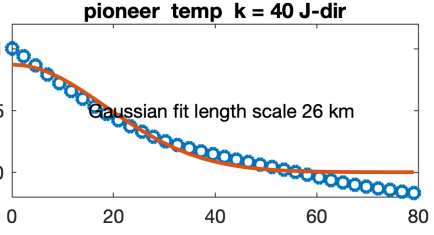






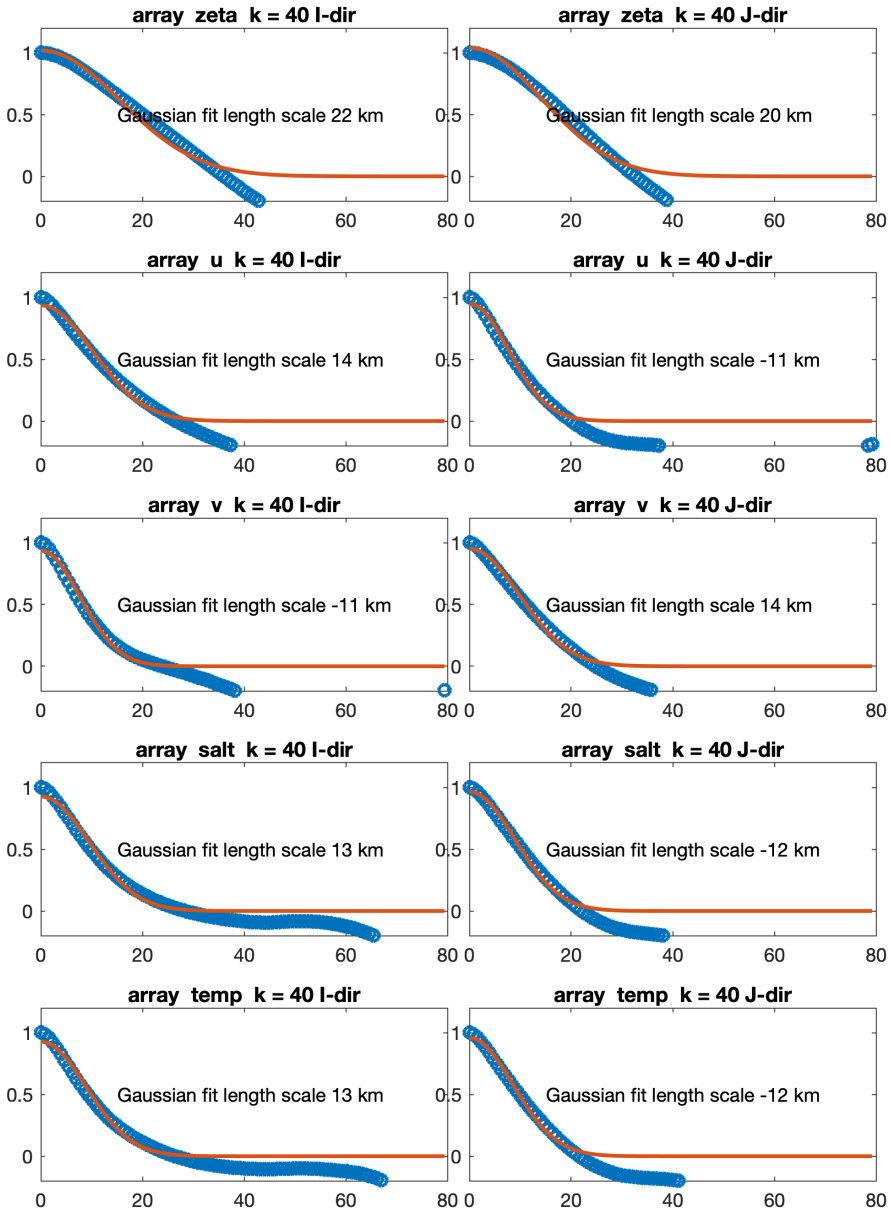






Fitted covariance functions for all variables

Array 700 m



Pioneer 2.2 km