# Regression kriging

Peter Finke, 2013..2017.

## 1. What does the program do?

Regression kriging is a program that can be used to map a numerical property (x) that is measured at known locations by using existing full-cover maps ($y_i$) with numerical or categorical data to construct regression models $xr=f(y_i)$. After application of the most suitable regression model, the model errors at the locations of x are interpolated by a spatial interpolation technique called kriging. The summation map (RKMap.asc) of the regression prediction and the model error is the final product of the program.

The table below summarizes the input and output files of the program.

| **Inputfiles** | **Description** | **Typical name** |
|---|---|---|
| maps | Ascii-grids | *.asc |
| Point data | Structured file | *.csv |
| **Outputfiles** | | |
| logfile | Report of all steps done in the program | RK.log |
| bitmaps | Maps and graphs of:<br>• Best fitted regression model<br>• CumulFreqDistribution of data<br>• CumulFreqDistribution of logit-transformed data<br>• Experimental Semivariogram<br>• Fitted variogram(s)<br>• Regression map<br>• Map of residuals<br>• Map regression kriging | <br>BestFit.bmp<br>DataCPDF.bmp<br>logit-DataCPDF.bmp<br>ExpSV.bmp<br>FittedSV*.bmp<br>RegressionMap.bmp<br>ResidualsMap.bmp<br>RKMap.bmp |
| Regressions | For every fitted regression model a file *.out<br>(also of the fitted variograms) | *.out |
| Kriging results | Kt3d*.*; residuals*.txt | |
| Temporary | *.txt | |
| maps | Ascii-grids of:<br>Regression<br>Residuals<br>Kriging variance<br>Regression Kriging<br>Total variance | <br>RegressionMap.asc<br>ResidualsMap.asc<br>ResidualsKrigingVariance.asc<br>RKMap.asc<br>RKTotalVariance.asc |

## 2. Data import

At startup, the following screen appears:

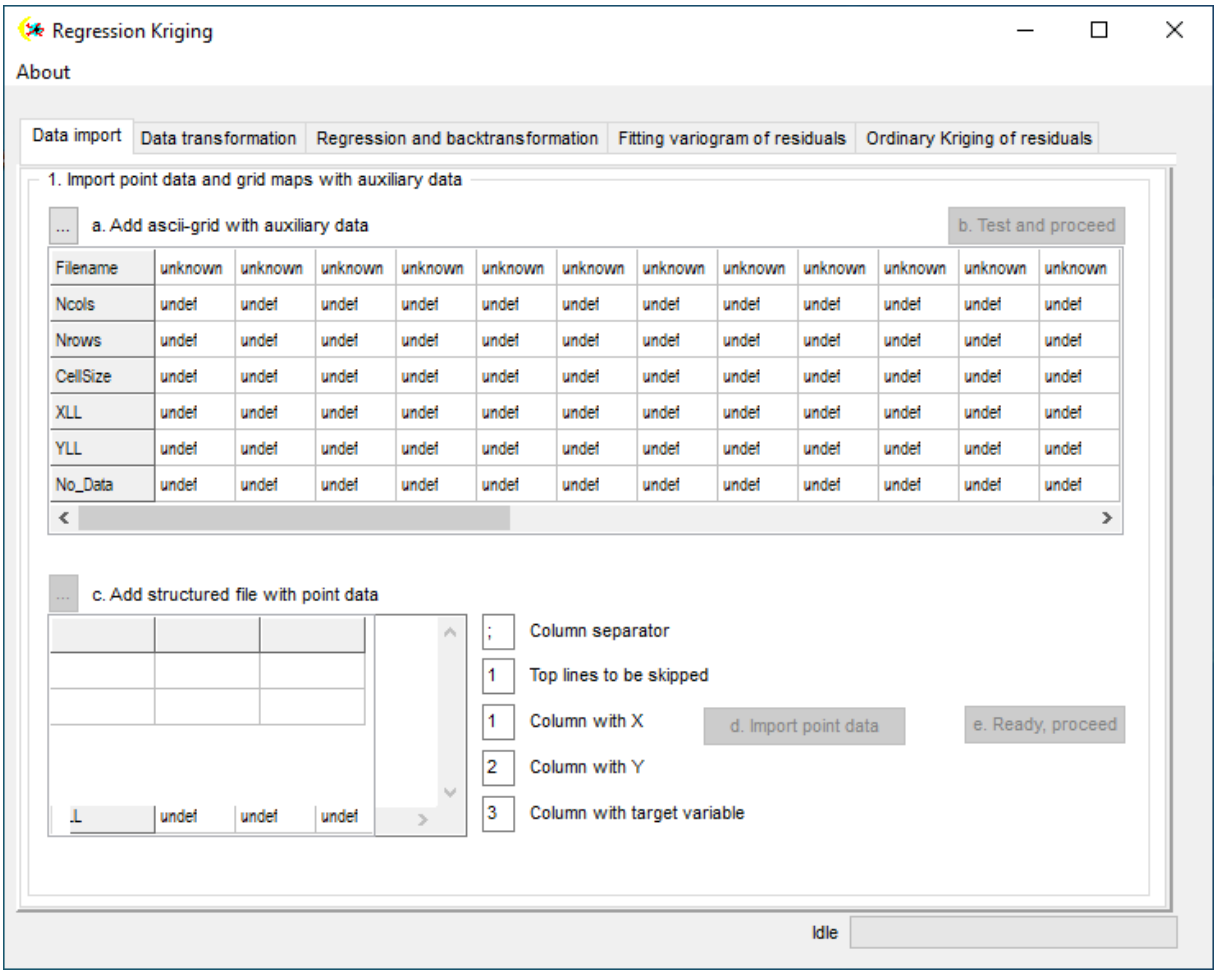


*Figure 1 Startscreen Regression Kriging software*

The working order is indicated by numbers and codes a, b, etc.

a. The first activity is to import grid data (ascii-grid maps, e.g. exported from ARC-GIS). Ascii grids are text-files as in the box below:

```
NCOLS 320
NROWS 420
XLLCORNER 100699.000000
YLLCORNER 205199.000000
CELLSIZE 5.000000
NODATA_VALUE -9999
0.0480 -0.0740 0.0600 -0.0600 0.0180 ...
...
-0.0340 0.1220 -0.0100 -0.0020 -0.0160
```

Where NCOLS is the number of X-axis grid cells, NROWS is the number of Y-axis grid cells, XLLCORNER, YLLCORNER give the coordinates of the lowest left cell in the grid, CELLSIZE informs about the size of the grid cell in the units of the coordinates, and NODATA_VALUE gives the code for blank values in the grid coverage. These data are named “grid extent” for short.

Below that we find for every row (Y) all values along the X-axis for all grid cells.

By pressing the button 1a (Figure 1) an ascii grid can be imported.

b. After all ascii grids are imported, button 1b is pressed and then the software checks if the grid extents are the same for all imported maps.
c. Pressing button 1c will allow importing the file with point data. Typically, this should be a comma (or semi-colon) separated file as exported from Excel. If the file has one or more header lines, these have to be skipped. Furthermore, the column numbers of X, Y and data columns must be given.
d. Pressing button 1d will read the point data and convert them to a readable table inside the program.

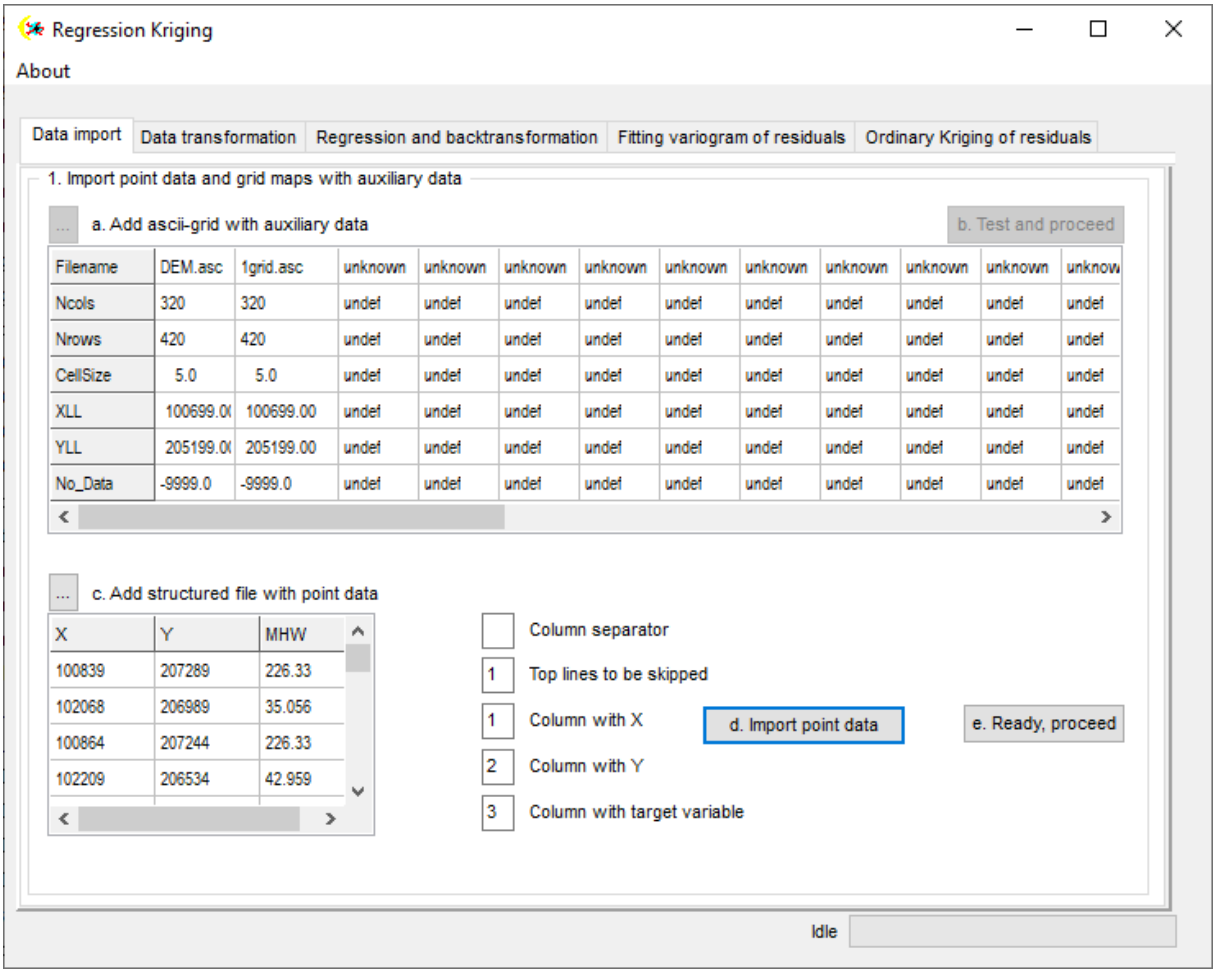


*Figure 2 Data import finished.*

e. Pressing button 1e will tell the program that the mapping can proceed.

## 3. Data transformation

A window opens such as in Figure 3. The upper part of the window allows to convert a categorical map (e.g., the texture class) into a set of indicator maps, e.g. one that only contains grid cells containing clay soils, one for sand, etcetera. It is assumed that the categorical map contains numbers and that the meaning of these numbers is known (e.g., 1=clay, 2=sand, …).

Hereto the categorical map must be selected and after pressing button 2b the identification of indicators can start.

The lower part of the window allows to work with a logit-transformation of the values at the point locations. The 2 figures show the data before (left) and after (right) a logit-transformation. Advise is

to choose the option in which the red (data) and blue (normal distribution) lines maximally coincide. In Figure 3, the checkbox  for logit transformation should best be unchecked.

Button 2f is pressed to proceed with the regression.

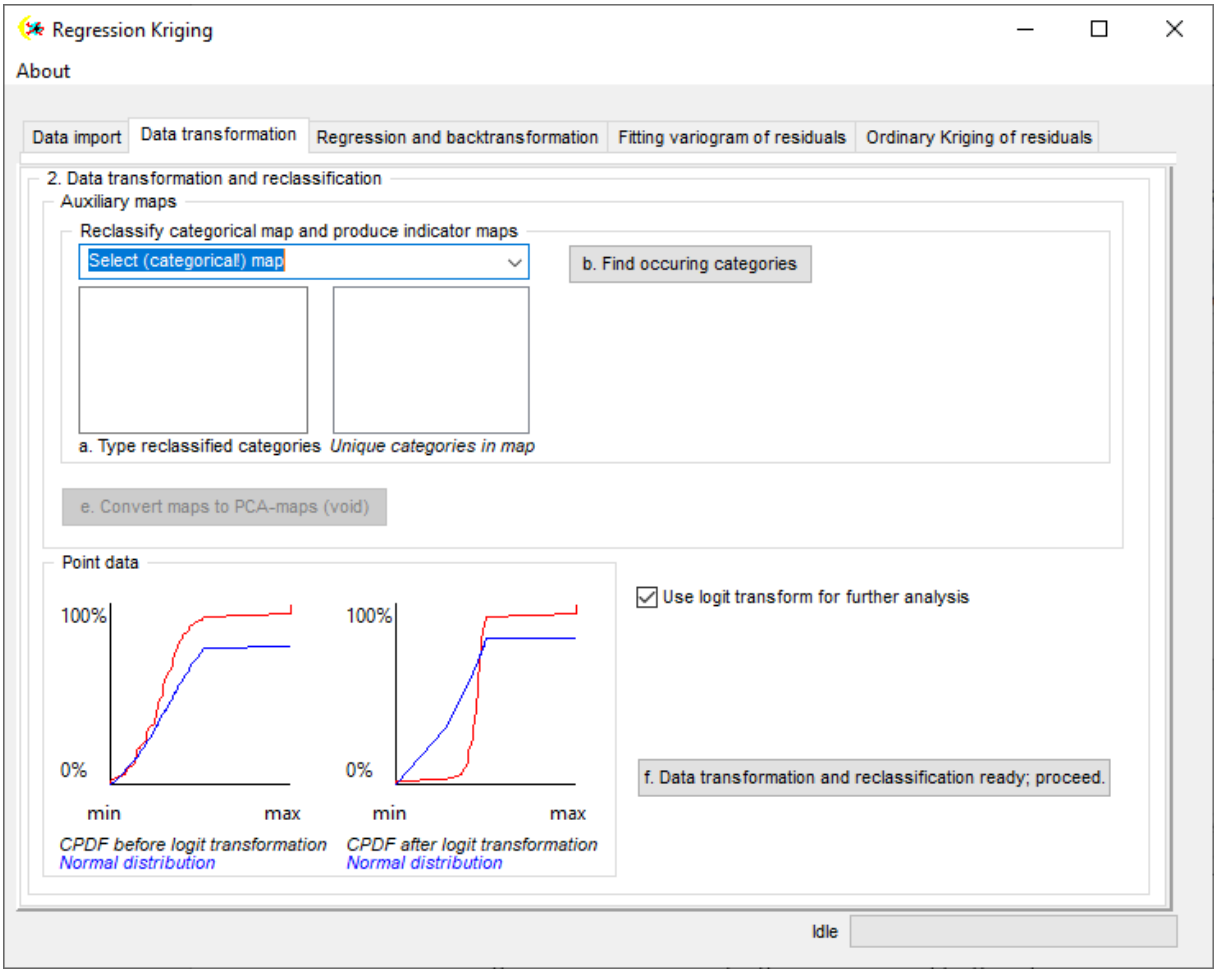


*Figure 3 Data transformation*

## 4. Regression and back-transformation

Pressing the button 3a will start the regression process. At the locations of the point data, values in the ascii-grids are looked up and all possible regression models are fit. The number of models that are fit depends on the number of ascii-grids selected in the data import window. Figure 4 gives the result after all regressions are done and the best model is selected by comparison of Mallow's Cp coefficients of all fitted models. The graph gives the relation between measured and fitted values at the point locations.

Pressing button 3b will then produce a map by applying the regression model onto the full map extent of the grid maps that are selected in the regression model.

Pressing button 3c is pressed to proceed to the next step: identification of the spatial structure of the errors in the regression map. These errors are called "residuals" below.

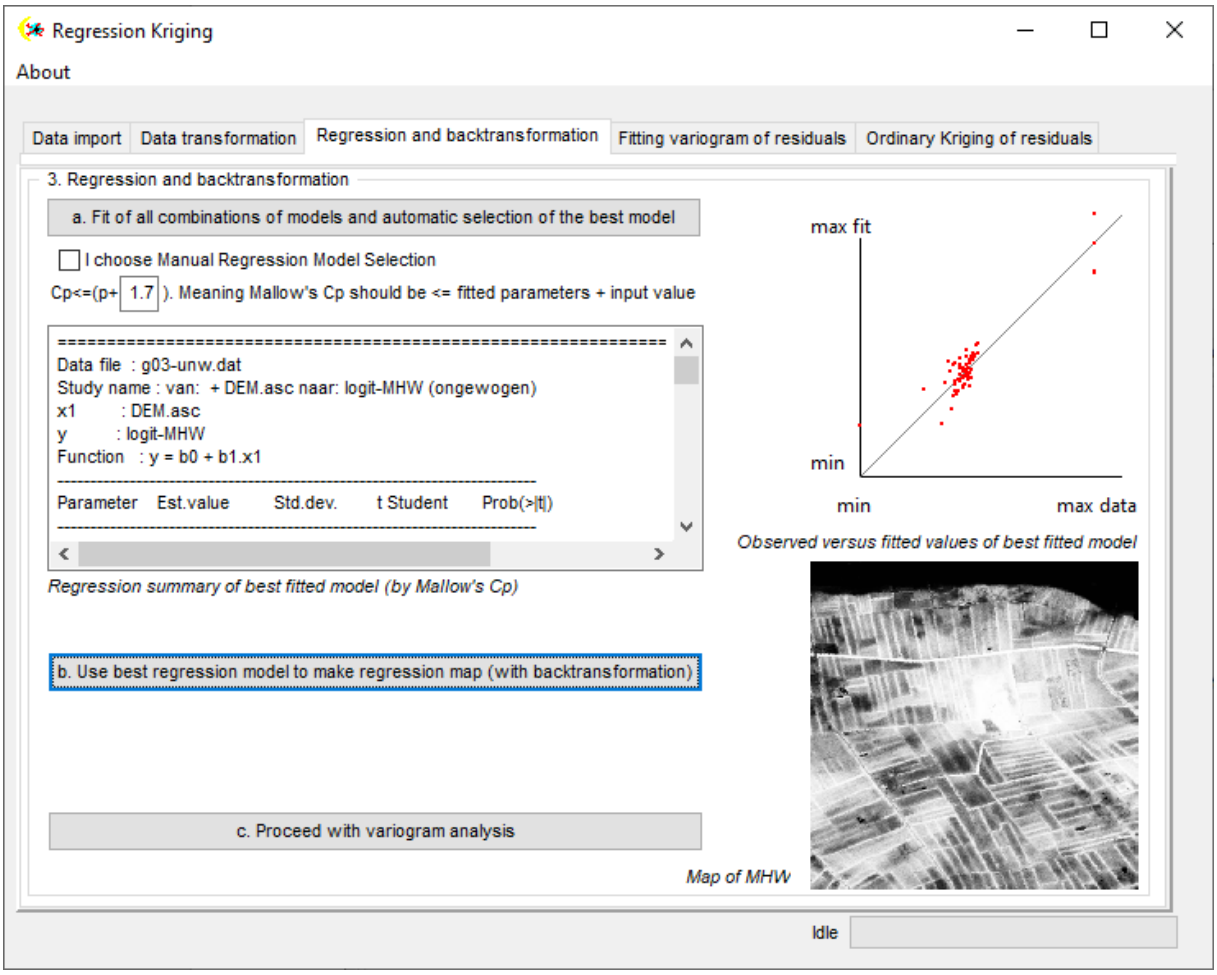


*Figure 4 Regression and back-transformation screen after the regression is done*

## 5. Fitting variogram of residuals

Fitting the variogram of the residuals is done in 2 steps:

a. The experimental variogram is calculated, using the point residuals. This involves (i) choosing a number of distance classes and (ii) choosing the size of the distance classes. The product of number x size should not be too large (i.e. be inside the area of interest). When this product is too high, the program will indicate this with a red text. As a rule of thumb, a reasonable number of distance classes is between 15-25. When these choices are made, button 4b: make experimental variogram, is pressed and the point cloud graph is produced.

b. Thereafter, a semivariogram model is fitted through this point cloud. There are 2 models (gaussian and exponential) available, for each one, the nugget (semivariance at lag of 0) can either be fitted or fixed to 0. Also, the fitting can be based on the individual weight of each point or by assigning equal weight to each point. Individual weights are based on the number of point pairs used to estimate each point in the experimental variograms. When these choices are made, button 4c can be pressed to see the resulting fitted variogram. These steps can be repeated if desired. When the results are satisfying, button 4b: continue is pressed, which will start the Kriging interpolation part of the program.

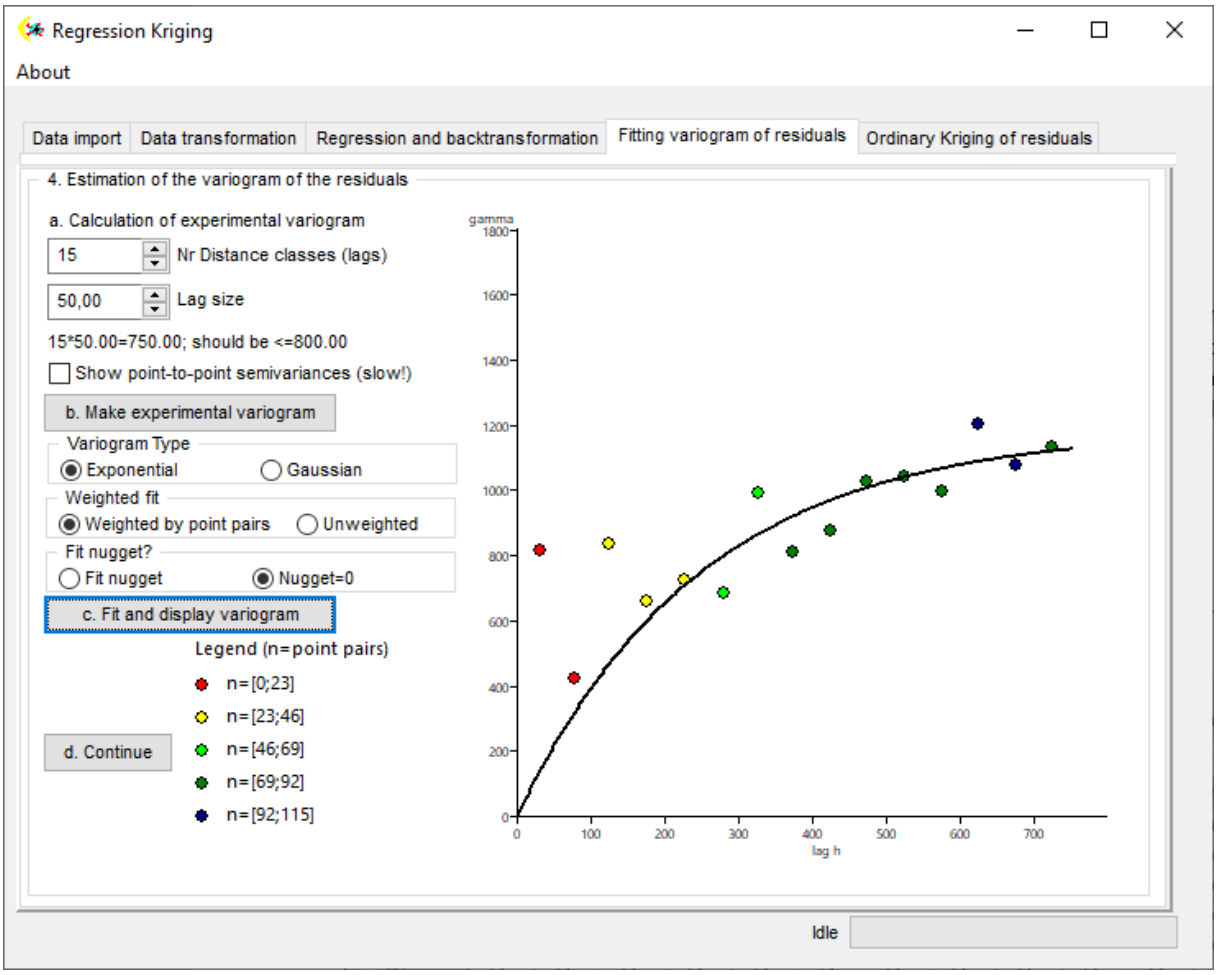


*Figure 5 Screen after fitting a variogram to the errors of the regression model.*

## 6. Ordinary Kriging of residuals

In this final step, the interpolation of the residuals is done and the regression map is added to the map of the residuals to give the final regression Kriging map.

The program will propose the last variogram model parameters chosen in the previous step.

Then the buttons 5b and 5c have to be pressed to prepare and perform the kriging. At the right part of the screen first the map of the residuals will be built. When this is ready, the map of the summation of the regression and the kriging-of-residuals will be displayed.

Then the program can be closed.

It will produce an outputfile RK.log summarizing all the steps taken.

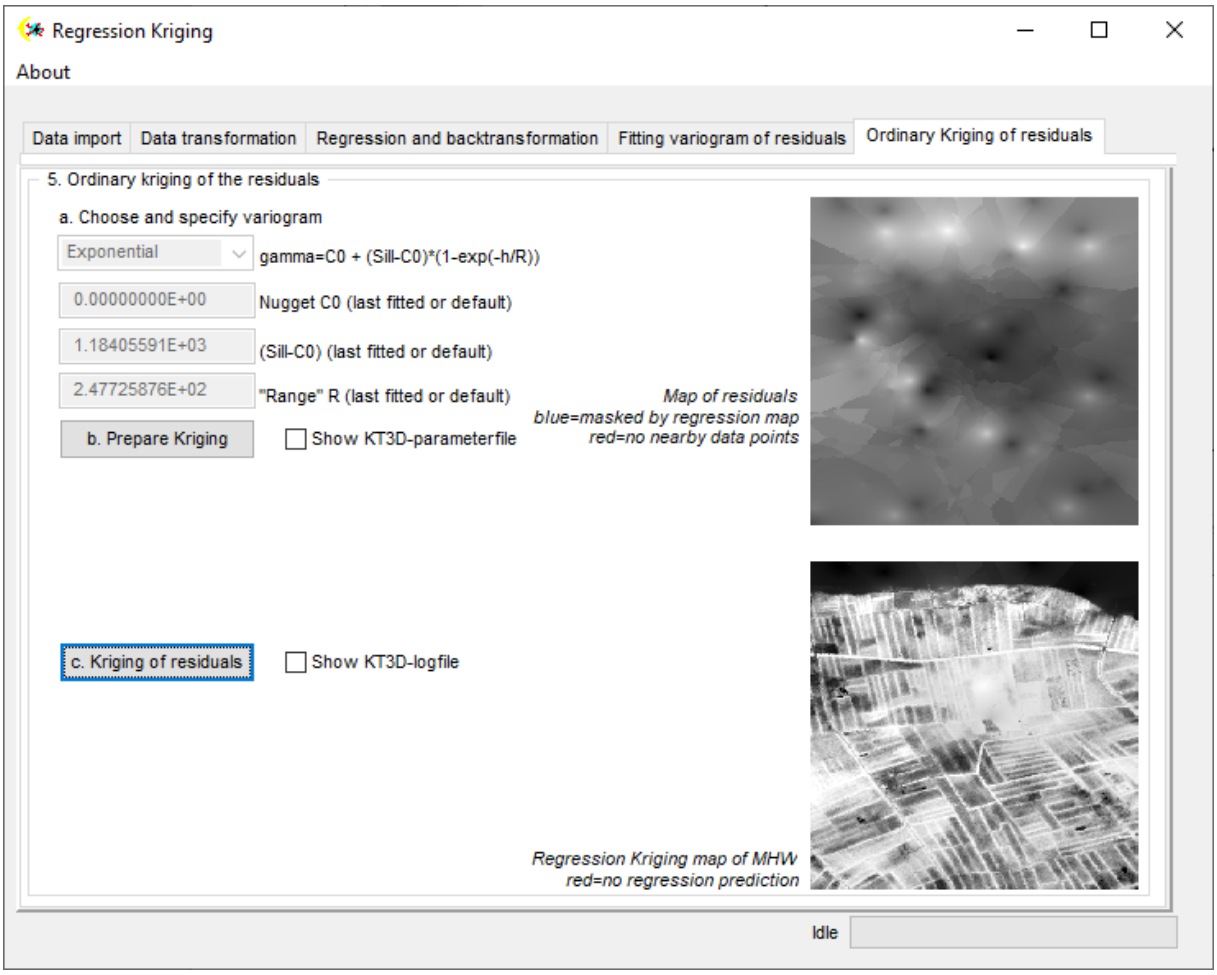


*Figure 6 Screen after regression kriging is finished.*