Contents

Usage of CRG_APPEND

Introducing the usage of crg_append. Examples are included for a set of common CRG-file formats. The file comments are optimized for the matlab publishing makro.

```
% NOTE
% One u-increment is used to adjust both crg-files into the right
% direction. Hence make sure you have a overlap by one (see examples).
%
%   Copyright 2005-2011 OpenCRG - VIRES Simulationstechnologie GmbH -
%   Holger Helmich
%
%   Licensed under the Apache License, Version 2.0 (the "License");
%   you may not use this file except in compliance with the License.
%   You may obtain a copy of the License at
%
%       http://www.apache.org/licenses/LICENSE-2.0
%
%   Unless required by applicable law or agreed to in writing, software
%   distributed under the License is distributed on an "AS IS" BASIS,
%   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
%   See the License for the specific language governing permissions and
%   limitations under the License.
%
%   More Information on OpenCRG open file formats and tools can be found at
%
%       http://www.opencrg.org
%
%   $Id: crg_test_append.m 1 2011-06-07 10:48:00Z hhelmich $
```

Test proceedings

Example 1 - 1.9:

- Load crg file (crg_read)
- Cut reference crg file (first crg_cut_iuiv call)
- Cut starting crg file patch (second crg_cut_iuiv call)
- Cut ending crg file patch (third crg_cut_iuiv call)
- Concatenate both patches (crg_append)
- Display result

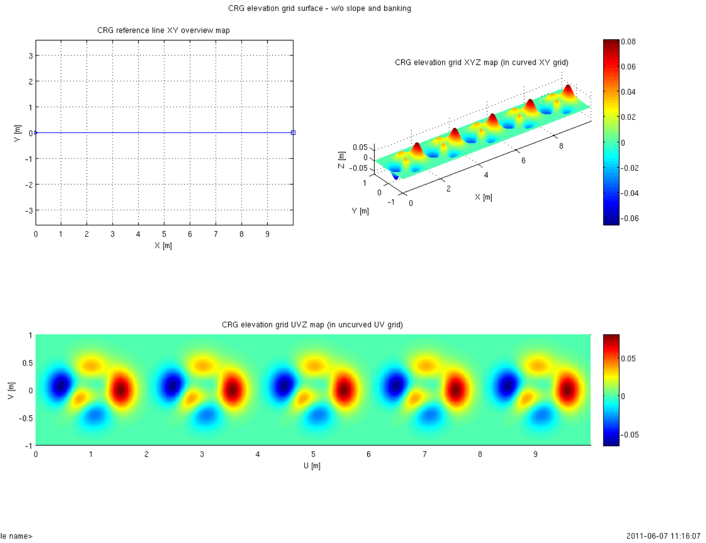
Example 2 - 10:

- Load different kind of crg files (crg_read)
- Concatenate both patches (crg_append)
- Display result

```
% DEFAULT SETTINGS
% clear enviroment
clear all;
close all;
```

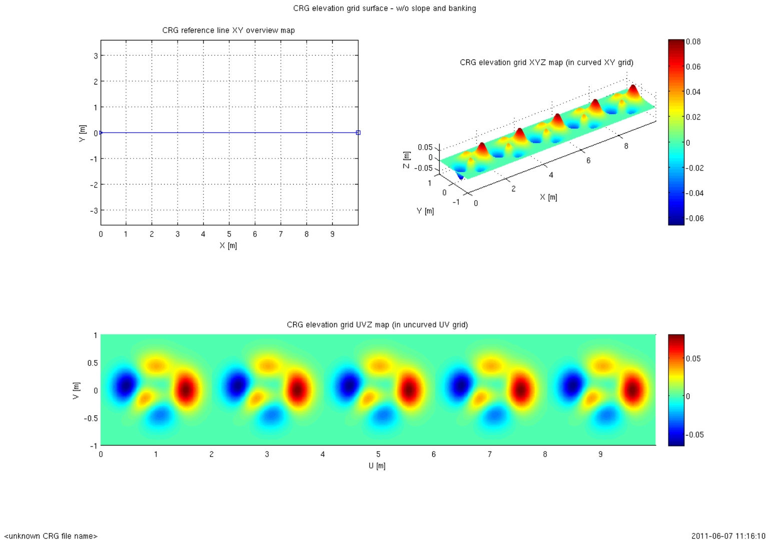
Test1 (defined by z matrix and scalar u and v specs)

```
data = crg_read('demo1.crg');

crg0 = crg_cut_iuiv(data, [1, 1000]);
crg1 = crg_cut_iuiv(crg0, [1, 501]);
crg2 = crg_cut_iuiv(crg0, [500, 1000]);
crg3 = crg_append(crg1, crg2);

crg_show_elgrid_surface(crg3, [1, 1000]);
```



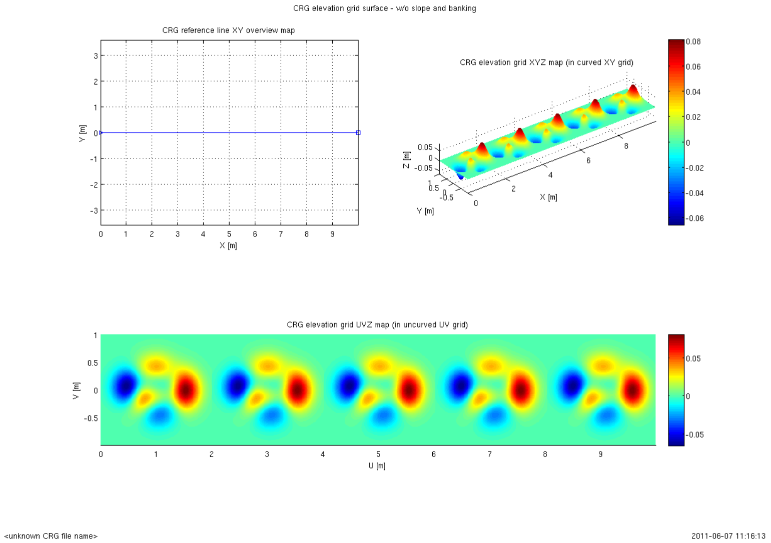
Test1.1 (. . . and evenly spaced v vector)

```
data = crg_read('demo2.crg');

crg0 = crg_cut_iuiv(data, [1, 1000]);
crg1 = crg_cut_iuiv(crg0, [1, 501]);
crg2 = crg_cut_iuiv(crg0, [500, 1000]);
crg3 = crg_append(crg1, crg2);

crg_show_elgrid_surface( crg3, [1, 1000] );
```



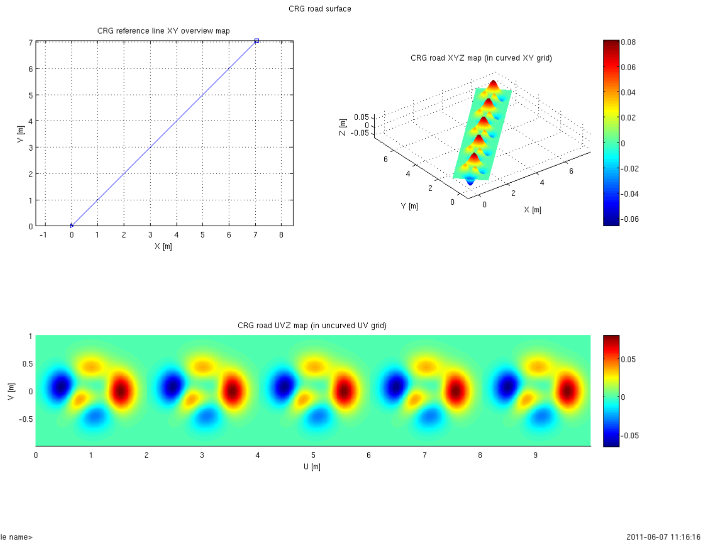
Test1.2 (. . . and unevenly spaced v vector)

```
data = crg_read('demo3.crg');

crg0 = crg_cut_iuiv(data, [1, 1000]);
crg1 = crg_cut_iuiv(crg0, [1, 501]);
crg2 = crg_cut_iuiv(crg0, [500, 1000]);
crg3 = crg_append(crg1, crg2);

crg_show_elgrid_surface( crg3, [1, 1000] );
```



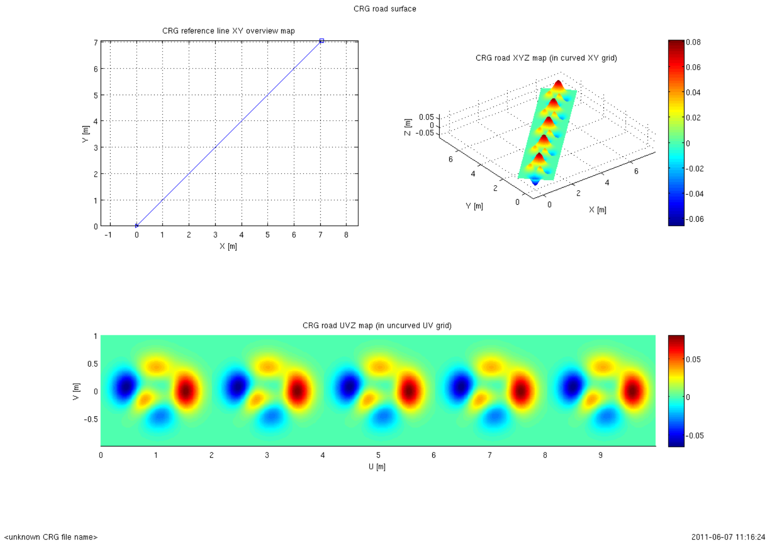
Test1.3 (. . . generate diagonal reference line by one p value)

```
data = crg_read('demo4.crg');

crg0 = crg_cut_iuiv(data, [1, 1000]);
crg1 = crg_cut_iuiv(crg0, [1, 501]);
crg2 = crg_cut_iuiv(crg0, [500, 1000]);
crg3 = crg_append(crg1, crg2);

crg_show_road_surface( crg3, [1, 1000] );
```



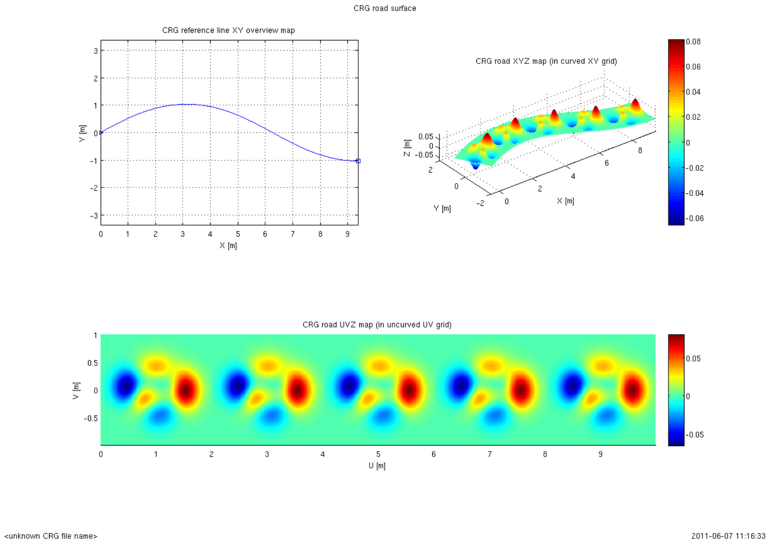
Test1.4 (. . . generate diagonal reference line by nu-1 p values)

```
data = crg_read('demo5.crg');

crg0 = crg_cut_iuiv(data, [1, 1000]);
crg1 = crg_cut_iuiv(crg0, [1, 501]);
crg2 = crg_cut_iuiv(crg0, [500, 1000]);
crg3 = crg_append(crg1, crg2);

crg_show_road_surface( crg3, [1, 1000] );
```



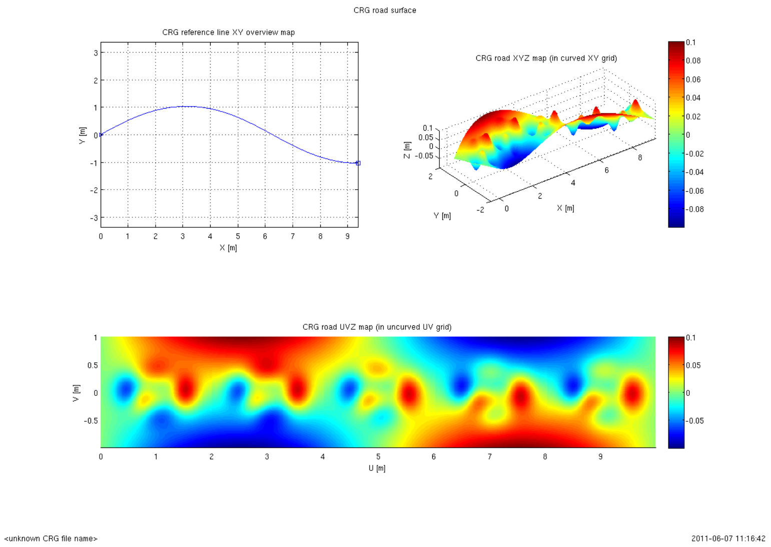
Test1.5 (. . . generate curved reference line)

```
data = crg_read('demo6.crg');

crg0 = crg_cut_iuiv(data, [1, 1000]);
crg1 = crg_cut_iuiv(crg0, [1, 501]);
crg2 = crg_cut_iuiv(crg0, [500, 1000]);
crg3 = crg_append(crg1, crg2);

crg_show_road_surface(crg3, [1 1000]);
```



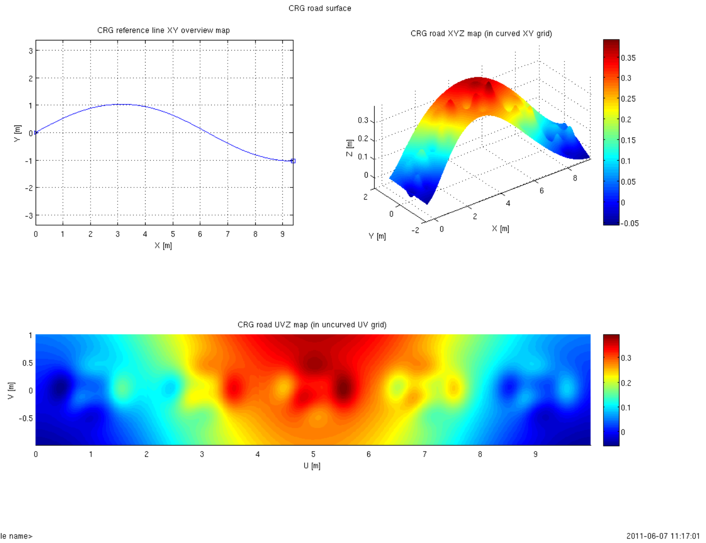
Test1.6 (. . . generate banking)

```
data = crg_read('demo7.crg');
crg0 = crg_cut_iuiv(data, [1, 1000]);
crg1 = crg_cut_iuiv(crg0, [1, 301]);
crg2 = crg_cut_iuiv(crg0, [300, 1000]);

crg3  = crg_append(crg1, crg2 );
crg_show_road_surface(crg3, [1 1000]);
```



Test1.7 (. . . generate slope)

```
data = crg_read('demo8.crg');

crg0 = crg_cut_iuiv(data, [1, 1000]);
crg1 = crg_cut_iuiv(crg0, [1, 301]);
crg2 = crg_cut_iuiv(crg0, [300, 1000]);

crg3 = crg_append(crg1, crg2 );
crg_show_road_surface(crg3, [1 1000]);
```


CRG road surface

CRG reference line XY overview map

CRG road XYZ map (in curved XY grid)

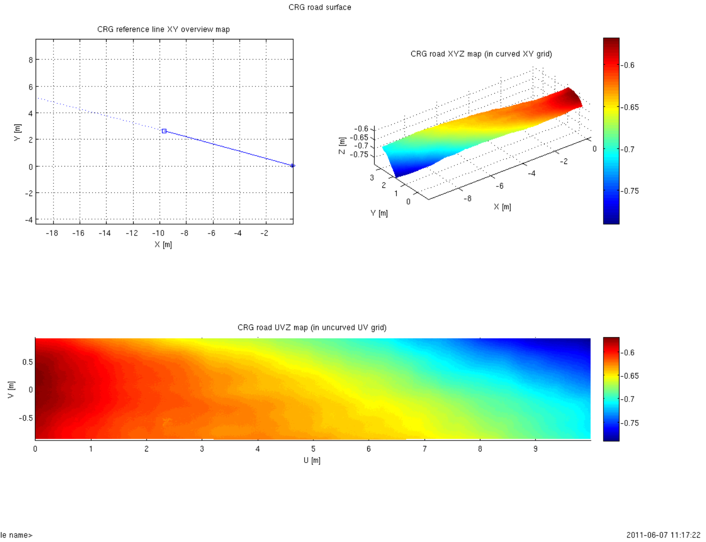
CRG road UVZ map (in uncurved UV grid)

<unknown CRG file name>

2011-06-07 11:17:01

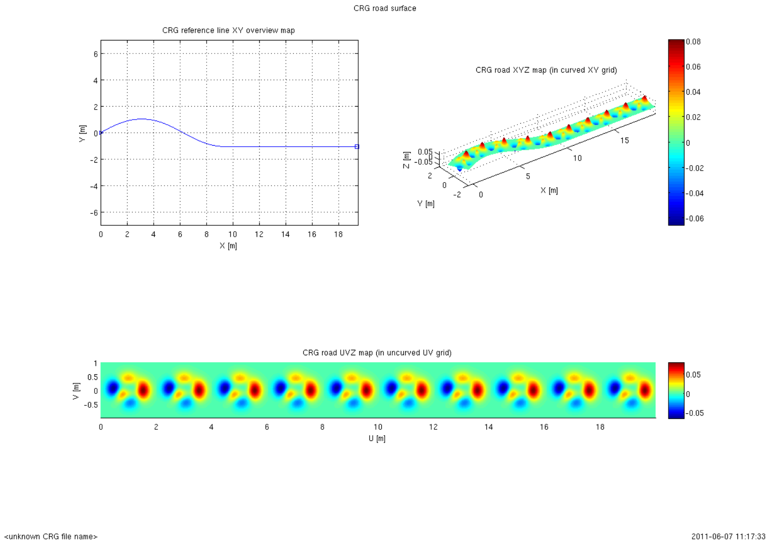
Test1.8 (+ WGS84)

```
crg = crg_read('../crg-bin/country_road.crg');

crg0 = crg_cut_iuiv(crg,  [1, 2000]);
crg1 = crg_cut_iuiv(crg0, [1, 1001]);
crg2 = crg_cut_iuiv(crg0, [1000, 2000]);

crg3  = crg_append(crg1, crg2 );

crg_show_road_surface(crg3, [1 1000]);
```



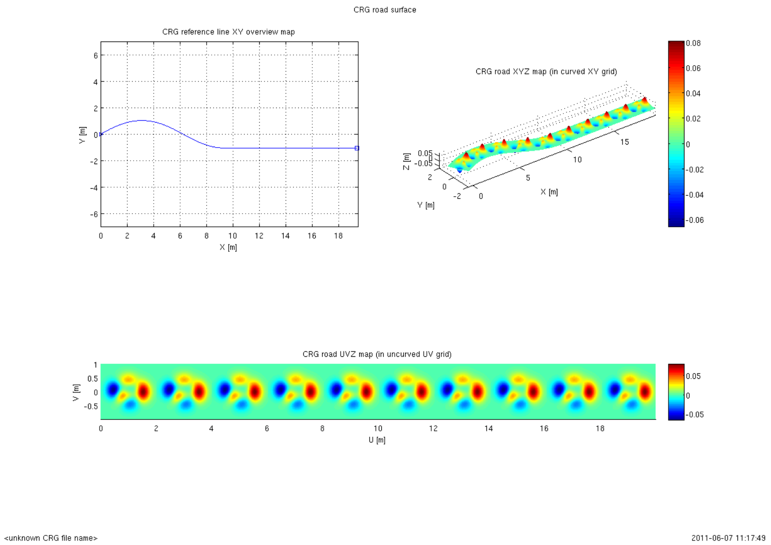
Test2 (curved crg + straight crg)

```
crg1 = crg_read('demo6.crg');
crg2 = crg_read('demo1.crg');
crg2 = crg_rerender(crg2, [], crg1.v);

crg3 = crg_append(crg1, crg2 );
crg_show_road_surface(crg3, [1 2000]);
```



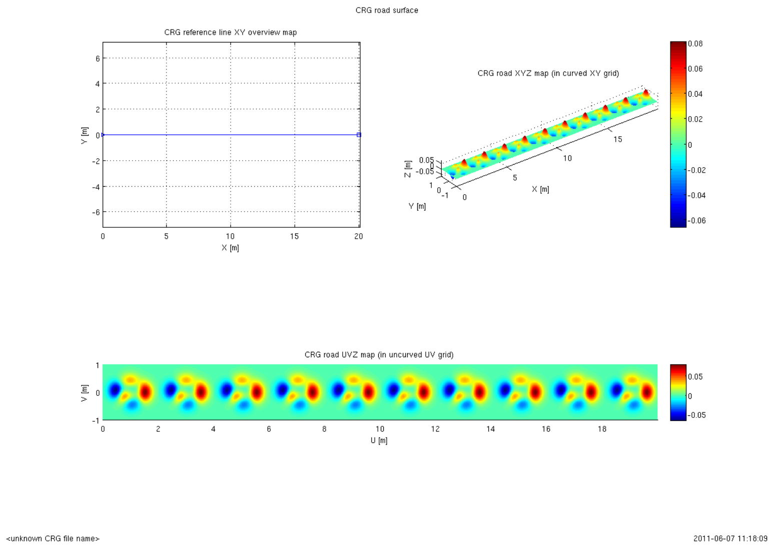
Test2.1 (curved crg + diagonal reference line)

```
crg1 = crg_read('demo6.crg');
crg2 = crg_read('demo3.crg');

crg3 = crg_append(crg1, crg2 );
crg_show_road_surface(crg3, [1 2000]);
```



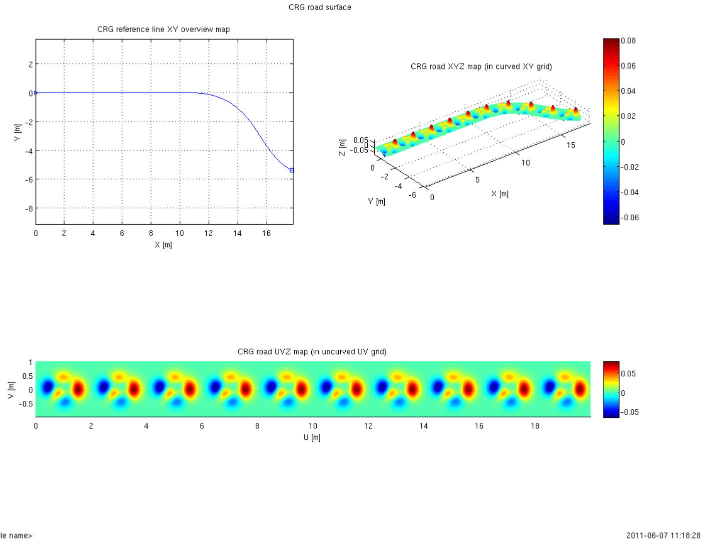
Test3 (straight crg + diagonal reference line)

```
crg1 = crg_read('demo1.crg');
crg2 = crg_read('demo3.crg');
crg2 = crg_rerender(crg2, [], crg1.v);

crg3 = crg_append(crg1, crg2 );
crg_show_road_surface(crg3, [1 2000]);
```



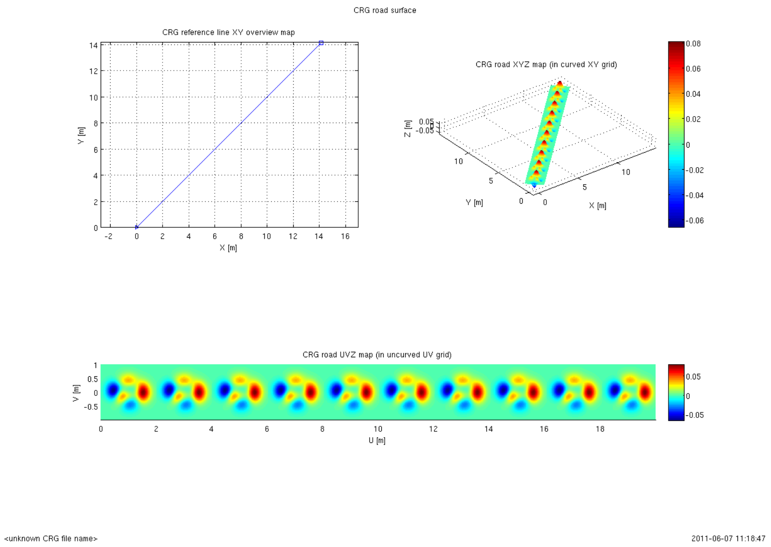
Test3.1 (straight crg + curved crg)

```
crg1 = crg_read('demo6.crg');
crg2 = crg_read('demo1.crg');
crg2 = crg_rerender(crg2, [], crg1.v);

crg3 = crg_append(crg2, crg1 );
crg_show_road_surface(crg3, [1 2000]);
```



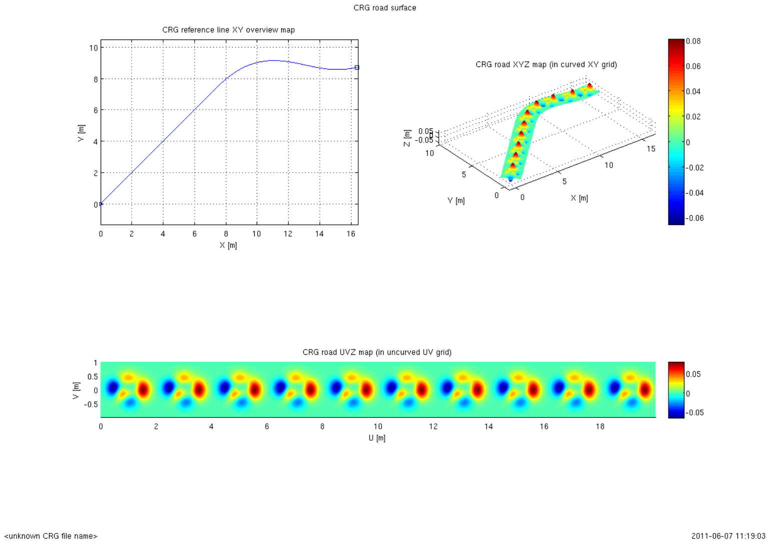
Test4 (diagonal reference line + straight crg)

```
crg1 = crg_read('demo4.crg');
crg2 = crg_read('demo1.crg');
crg2 = crg_rerender(crg2, [], crg1.v);

crg3 = crg_append(crg1, crg2);
crg_show_road_surface(crg3, [1 2000]);
```



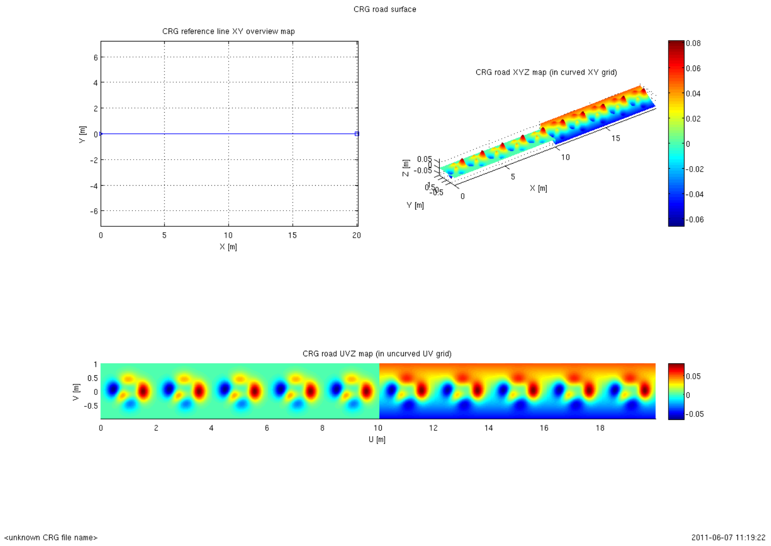
Test4.1 (diagonal reference line + curved crg)

```
crg1 = crg_read('demo6.crg');
crg2 = crg_read('demo4.crg');

crg3 = crg_append(crg2, crg1);
crg_show_road_surface(crg3, [1 2000]);
```



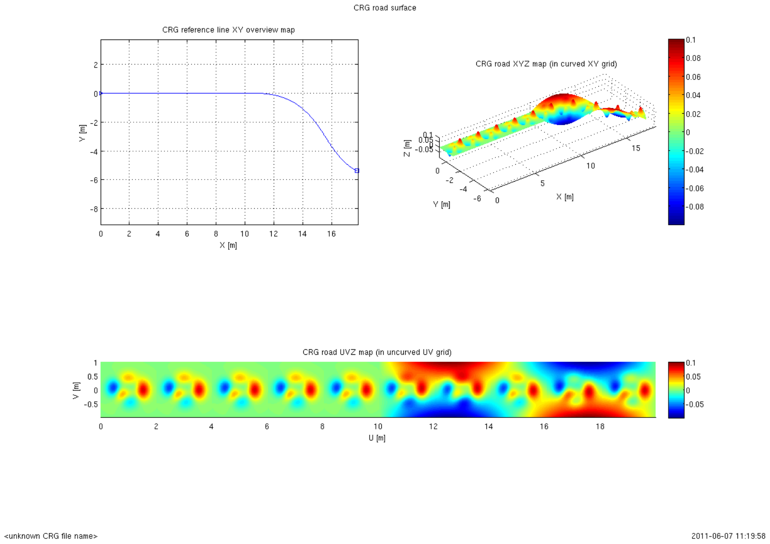
Test5 (no banking + constant banking)

```
crg1 = crg_read('demo3.crg');
crg2 = crg_read('demo1.crg');
crg2 = crg_rerender(crg2, [], crg1.v);

crg2.b = 0.05;
crg2.head.bbeg = 0.05;
crg2.head.bend = 0.05;
crg2 = crg_check(crg2);

crg3 = crg_append(crg1, crg2);
crg_show_road_surface(crg3, [1 2000]);
```



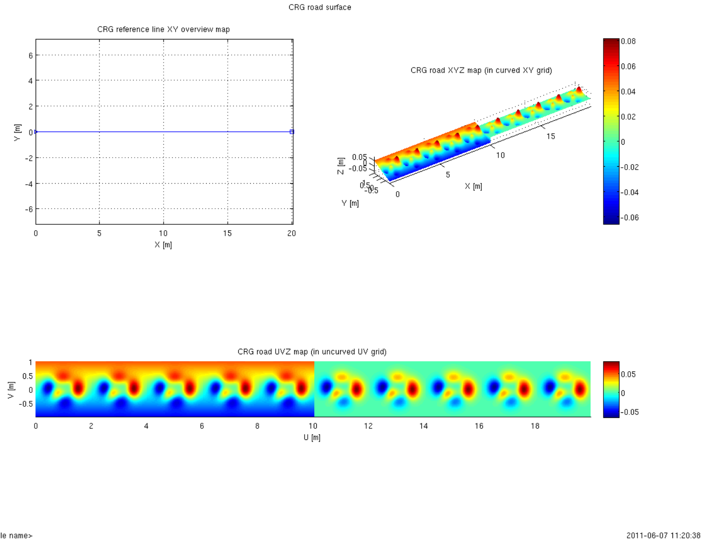
Test5.1 (no banking + variable banking)

```
crg1 = crg_read('demo3.crg');
crg2 = crg_read('demo7.crg');

crg3 = crg_append(crg1, crg2);
crg_show_road_surface(crg3, [1 2000]);
```



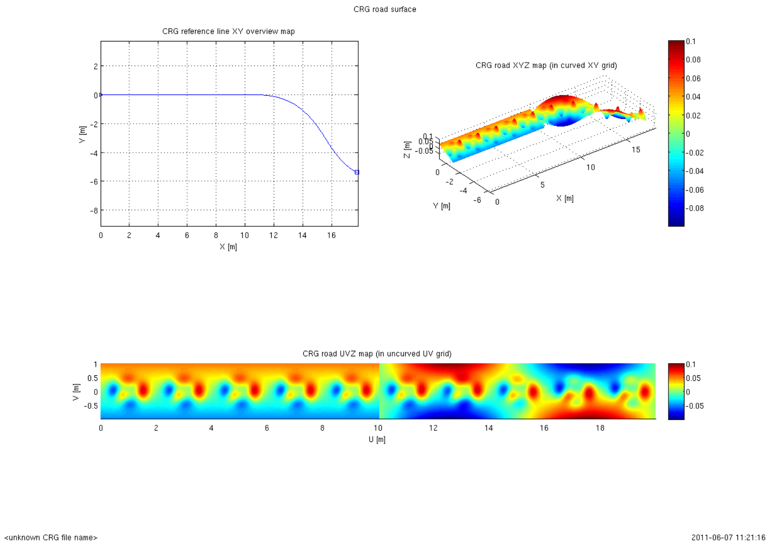
Test6 (constant banking + no banking)

```
crg1 = crg_read('demo3.crg');
crg2 = crg_read('demo1.crg');
crg2 = crg_rerender(crg2, [], crg1.v);

crg2.b = 0.05;
crg2.head.bbeg = 0.05;
crg2.head.bend = 0.05;
crg2 = crg_check(crg2);

crg3 = crg_append(crg2, crg1);
crg_show_road_surface(crg3, [1 2000]);
```



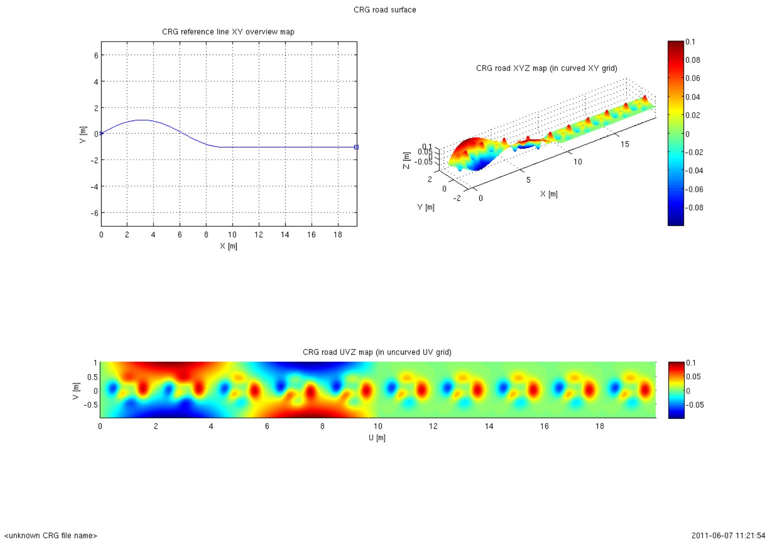
Test6.1 (constant banking + variable banking)

```
crg1 = crg_read('demo7.crg');
crg2 = crg_read('demo1.crg');
crg2 = crg_rerender(crg2, [], crg1.v);

crg2.b = 0.05;
crg2.head.bbeg = 0.05;
crg2.head.bend = 0.05;
crg2 = crg_check(crg2);

crg3 = crg_append(crg2, crg1);
crg_show_road_surface(crg3, [1 2000]);
```



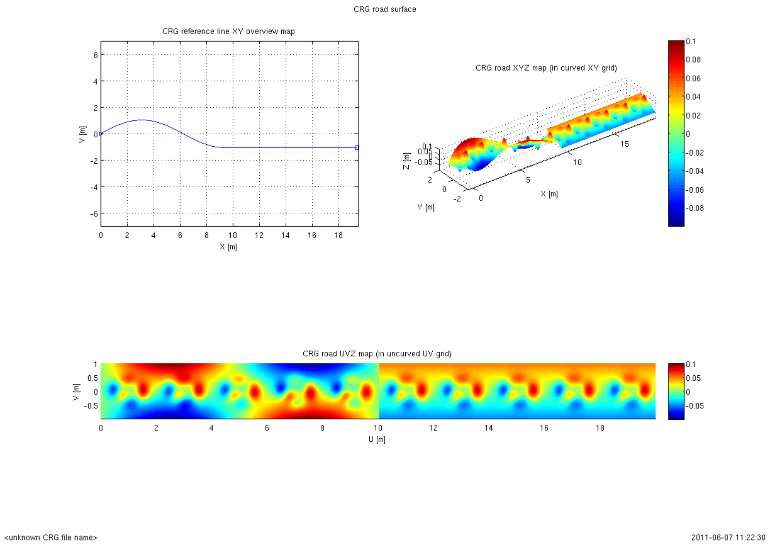
Test7 (variable banking + no banking)

```
crg1 = crg_read('demo3.crg');
crg2 = crg_read('demo7.crg');

crg3 = crg_append(crg2, crg1);
crg_show_road_surface(crg3, [1 2000]);
```



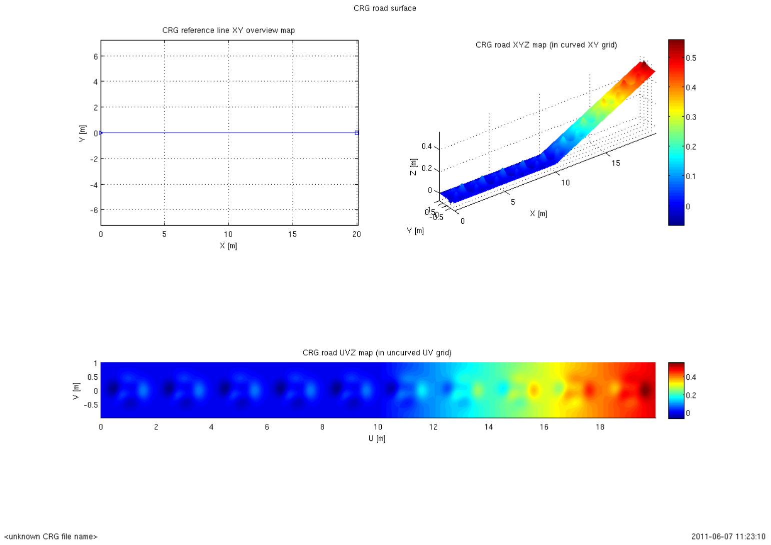
Test7.1 (variable banking + constant banking)

```
crg1 = crg_read('demo7.crg');
crg2 = crg_read('demo3.crg');

crg2.b = 0.05;
crg2.head.bbeg = 0.05;
crg2.head.bend = 0.05;
crg2 = crg_check(crg2);

crg3 = crg_append(crg1, crg2);
crg_show_road_surface(crg3, [1 2000]);
```



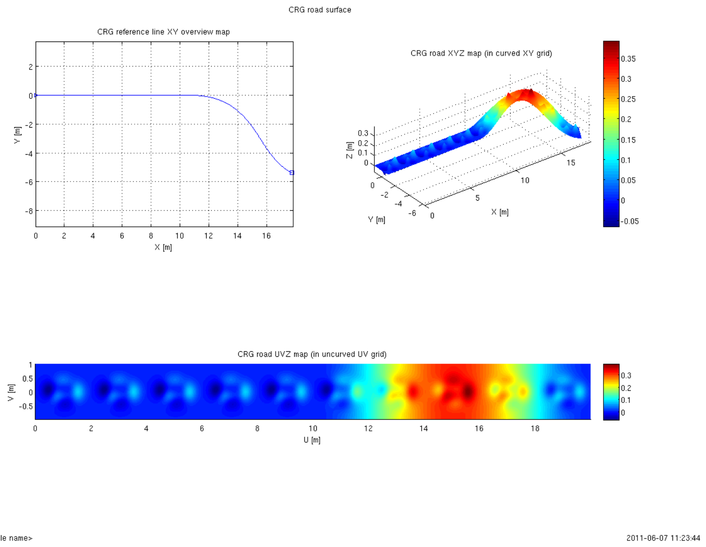
Test8 (no slope + constant slope)

```
crg1 = crg_read('demo3.crg');
crg2 = crg_read('demo1.crg');
crg2 = crg_rerender(crg2, [], crg1.v);

crg2.s = 0.05;
crg2.head = rmfield(crg2.head, 'send');
crg2.head = rmfield(crg2.head, 'sbeg');
crg2.head = rmfield(crg2.head, 'zbeg');
crg2.head = rmfield(crg2.head, 'zend');
crg2 = crg_check(crg2);

crg3 = crg_append(crg1, crg2);
crg_show_road_surface(crg3, [1 2000]);
```



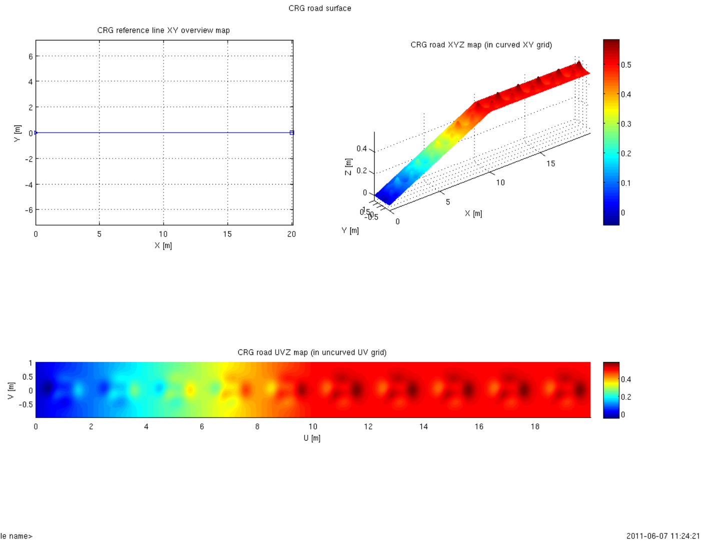
Test8.1 (no slope + variable slope)

```
crg1 = crg_read('demo3.crg');
crg2 = crg_read('demo9.crg');

crg3 = crg_append(crg1, crg2);
crg_show_road_surface(crg3, [1 2000]);
```



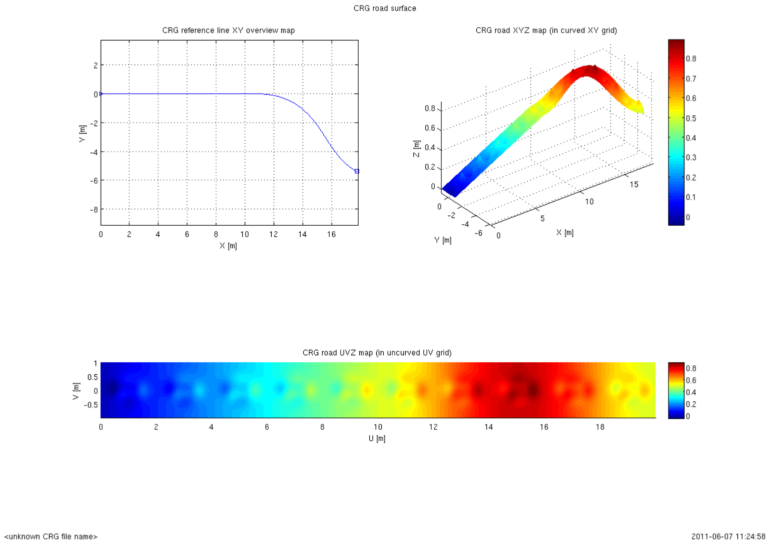
Test9 (constant slope + no slope)

```
crg1 = crg_read('demo3.crg');
crg2 = crg_read('demo1.crg');
crg2 = crg_rerender(crg2, [], crg1.v);

crg2.s = 0.05;
crg2.head = rmfield(crg2.head, 'send');
crg2.head = rmfield(crg2.head, 'sbeg');
crg2.head = rmfield(crg2.head, 'zbeg');
crg2.head = rmfield(crg2.head, 'zend');
crg2 = crg_check(crg2);

crg3 = crg_append(crg2, crg1);
crg_show_road_surface(crg3, [1 2000]);
```



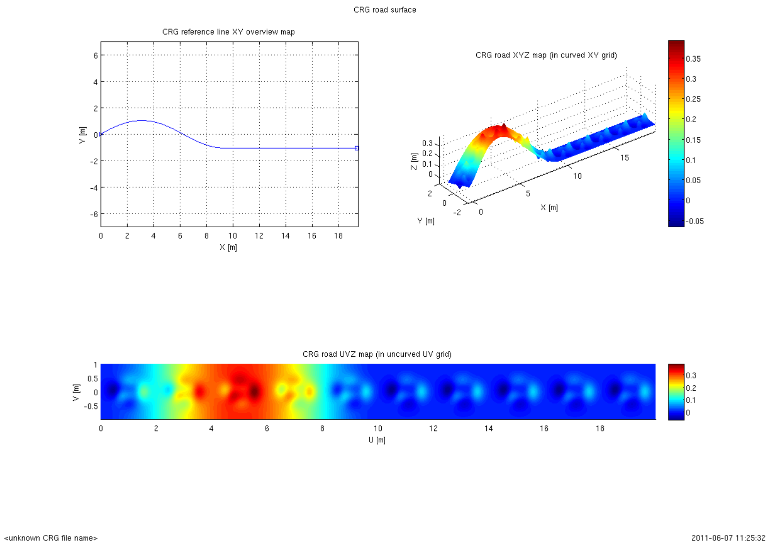
Test9.1 (constant slope + variable slope)

```
crg1 = crg_read('demo9.crg');
crg2 = crg_read('demo1.crg');
crg2 = crg_rerender(crg2, [], crg1.v);

crg2.s = 0.05;
crg2.head = rmfield(crg2.head, 'send');
crg2.head = rmfield(crg2.head, 'sbeg');
crg2.head = rmfield(crg2.head, 'zbeg');
crg2.head = rmfield(crg2.head, 'zend');
crg2 = crg_check(crg2);

crg3 = crg_append(crg2, crg1);
crg_show_road_surface(crg3, [1 2000]);
```



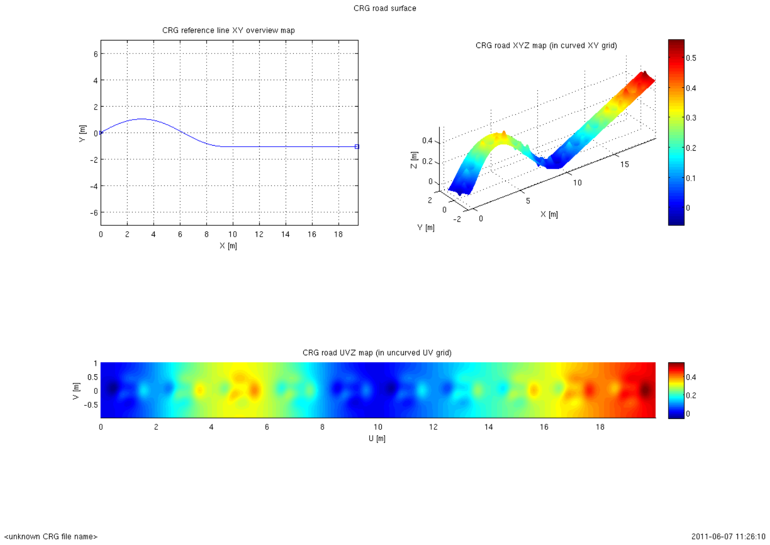
Test10 (variable slope + no slope)

```
crg1 = crg_read('demo3.crg');
crg2 = crg_read('demo9.crg');

crg3 = crg_append(crg2, crg1);
crg_show_road_surface(crg3, [1 2000]);
```



Test10.1 (variable slope + constant slope)

```
crg1 = crg_read('demo9.crg');
crg2 = crg_read('demo1.crg');
crg2 = crg_rerender(crg2, [], crg1.v);

crg2.s = 0.05;
crg2.head = rmfield(crg2.head, 'send');
crg2.head = rmfield(crg2.head, 'sbeg');
crg2.head = rmfield(crg2.head, 'zbeg');
crg2.head = rmfield(crg2.head, 'zend');
crg2 = crg_check(crg2);

crg3 = crg_append(crg1, crg2);
crg_show_road_surface(crg3, [1 2000]);
```


CRG road surface
CRG reference line XY overview map
Y [m]
X [m]
CRG road XYZ map (in curved XY grid)
Z [m]
Y [m]
X [m]
CRG road UVZ map (in uncurved UV grid)
V [m]
U [m]
<unknown CRG file name>
2011-06-07 11:26:10