

SPATIAL ANALYSIS PERFORMED BASED ON THE NUMERICAL MODEL OF THE ELEVATION USING AUTODESK MAP PLATFORM

Constantin Bofu, Assoc.Prof.PhD.eng., – TUIasi, Faculty of Hydrotechnics, Geodesy and Environmental Engineering, constantinbofu@yahoo.com

Costel Boariu, Lecturer PhD.eng– TUasi, Faculty of Hydrotechnics, Geodesy and Environmental Engineering , costelboariu@gmail.com

Maximilian Diac, As. PhD.eng– TUIasi, Faculty of Hydrotechnics, Geodesy and Environmental Engineering , maximiliandiac@yahoo.com

Abstract: *To model the rain/drain process on GIS platforms it is involved an automated technique that enables the generation of numerical models of the elevation that are able to simulate this operation. When the GIS platform is not addressed to complete such a rain/drain model phenomenon it is needed to develop a specific application which will allow parameter determinations and thematic zoning. The following study is based on a created numerical model of altimetry data, on which are developed applications customized to generate thematic zoning of calculus draining parameters.*

Keywords: *GIS, model, sequences , parameters, morphometric*

1. Introduction

In the process of simulation the rain/drain phenomenon on numerical models of the field it is necessary to determinate some morphometric characteristics of the territory such as the slope identified in each point of the grid and the drainage parameters of the system: directions of flow, accumulations points of flow, concentration sequences, alpha coefficients. Because the simulation is based on a grid model with a unit of 10 meters and a number of 264x285 cells it is used the notion of matrix for calculating these parameters. These parameters set are determined using Autodesk MAP 3D platform and AutoLISP and C++ programming languages. The paper addresses the determination of the morphometrical parameter slope and its zoning on intervals.

2. MATERIAL AND METHODS

2.1. MATERIALS USED

The two data sources that were involved in achieving this study are the plan sheet with a scale of 1: 5000 , that generated the 3D elevation model (Figure no.2.1), and all the data obtained from the meteorological stations in the vicinity of the study area (Table 2.1)

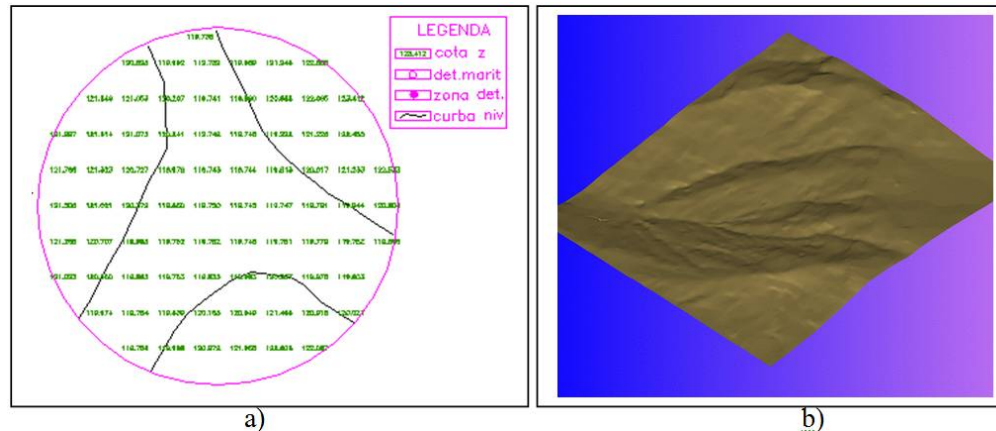


Figure 2.1 Attribute and graphic data a) Matrix of slopes b) Digital model of the terrain [1]

2.2. WORKING METHOD

Realizarea aplicației pentru determinarea pantelor

To determinate the slope in each point of the grid there was used the gradient of an area expression, formula which has the following form [1]:

$$g = (g_x, g_y, g_z) = \bar{V} \{z - Z(x, y)\} = \left\{ -\frac{dZ}{dx}, -\frac{dZ}{dy}, 1 \right\}$$

For a matrix of dimensions (i*j) there are:

$$g_{xij} = -\frac{Z_{i+1,j} - Z_{i-1,j}}{2d}; \quad g_{yij} = -\frac{Z_{i,j+1} - Z_{i,j-1}}{2d};$$

$$g_{zij} = -1.$$

unde: Z - the level of points;

d - the dimension of the grid side or of the diagonal.

Using the geometric equation of the gradient it is determined the slope value in each point (i,j):

$$P_{i,j} = \sqrt{g_x^2 + g_y^2}$$

The following 111 lines described below represent the written program in C++:

```

1. #include <stdlib.h>
2. #include <fstream.h>
3. #include <math.h>
4. #include <string.h>
5. #include <iostream.h>
6. #include <except.h>
7. #include <iomanip.h>
8. void scriere(long &x,long &y, double &z,double &p, ofstream &outs)
9. {
10. const sx=7, sy=8, sz=8, sp=7;
11. char bufx[sx], bufy[sy], bufz[sz], bufp[sp], lf[1]="\n";
12. ltoa(x, bufx, 10);
13. ltoa(y, bufy, 10);

```

```
14.  gcvt(z, 8, bufz);
15.  gcvt(p, 6, bufp);
16.  cout << bufx << " " << bufy <<ios::fixed<<setprecision(3)<< " " <<
    bufz<<" " <<bufp<<endl;
17.  outs.write(bufx,6);
18.  outs.write(" ",1);
19.  outs.write(bufy,7);
20.  outs.write(" ",1);
21.  outs.write(bufz,7);
22.  outs.write(" ",1);
23.  outs.write(bufp,6);
24.  outs.write(lf,1);
25.  }
26.  void citire(double &z, ifstream &ins)
27.  {
28.  const bsx = 8, bsy = 9, bsz = 8;
29.  char bufx[bsx], bufy[bsy], bufz[bsz], c;
30.  long x,y;
31.  ins.get(bufx, bsx);
32.  x = atoi(bufx);
33.  cout << x << endl;
34.  ins.get(bufy, bsy);
35.  y = atoi(bufy);
36.  cout << y << endl;
37.  ins.get(bufz, bsz);
38.  z = atof(bufz);
39.  cout << z << endl;
40.  ins.get(c);
41.  }
42.  void main()
43.  {
44.  ifstream ins("C:\\bc5\\Test\\mncote5000.txt");
45.  if(!ins)
46.  {
47.  cerr << "Nu pot deschide fisierul: mncote5000.txt";
48.  exit(1);
49.  }
50.  ofstream outs("C:\\bc5\\Test\\mnpanta.txt",ios::app);
51.  if(!outs)
52.  {
53.  cerr << "Nu pot deschide fisierul: mnpanta.txt ";
54.  exit(1);
55.  }
56.  int i, j, m=265, n=284;
57.  long x,y;
58.  double z, Gx, Gy;
59.  double **tabz, **tabp;
60.  try {
61.  tabz = new double *[m];
```

```
62.  tabp = new double *[m];
63.  for (int j=0; j<m; j++)
64.  { tabz[j] = new double [n];
65.  tabp[j] = new double [n]; }
66.  }
67.  catch (xalloc)
68.  {
69.  cout << "NU POT ALOCA. PARASESC!";
70.  exit(-1);
71.  }
72.  for (i=0; i<m; i++)
73.  for (j=0; j<n; j++)
74.  {
75.  citire(z,ins);
76.  tabz[i][j]=z;
77.  tabp[i][j]=0;
78.  }
79.  for (i=0; i<m; i++)
80.  for (j=0; j<n; j++)
81.  {
82.  if (i==0) { Gy=(tabz[0][j]-tabz[1][j])/10;
83.  tabp[0][j]=Gy*Gy+tabp[0][j];}
84.  if (i==264) { Gy=(tabz[263][j]-tabz[264][j])/10;
85.  tabp[264][j]=Gy*Gy+tabp[264][j];}
86.  if (i!=0&& i!=264) { Gy=(tabz[i-1][j]-tabz[i+1][j])/20;
87.  tabp[i][j]=Gy*Gy+tabp[i][j];}
88.  }
89.  for (i=0; i<m; i++)
90.  for (j=0; j<n; j++)
91.  {
92.  if (j==0) { Gx=(tabz[i][0]-tabz[i][1])/10;
93.  tabp[i][0]=Gx*Gx+tabp[i][0];}
94.  if (j==283) { Gx=(tabz[i][282]-tabz[i][283])/10;
95.  tabp[i][283]=Gx*Gx+tabp[i][283];}
96.  if (j!=0&& j!=283) { Gx=(tabz[i][j-1]-tabz[i][j+1])/20;
97.  tabp[i][j]=Gx*Gx+tabp[i][j];}
98.  }
99.  for (i=0; i<m; i++)
100. for (j=0; j<n; j++)
101. {
102. tabp[i][j]=100*sqrt(tabp[i][j]);
103. cout << tabp[i][j] << " ";
104. }
105. for (y=5210210; y<=5212850; y=y+10)
106. for (x=544350; x<=547180; x=x+10)
107. {
108. i=(y-5210210)/10;
109. j=(x-544350)/10;
110. scriere(x,y,tabz[i][j],tabp[i][j],outs);
```

```

111. }
112. for (int i=0; i<m; i++)
113. { delete[] tabz[i];
114. delete[] tabp[i];}
115. delete[] tabz;
116. delete[] tabp;
117. }
    
```

All the slope values determined are entered in the matrix file specified "MNPANTA.TXT", containing the x, y values of each point of the grid as well as the slope value attribute of them.

3. RESULTS

3.1. The map of the slope values grid for the study area

The graphic map with the information about the slope (displayed in each point of the grid associated with the numerical model of the field) was done through a Lisp application, based on the slope matrix "MNPANTA.TXT" (Figure 3.1). The process will be dealt with on another occasion.

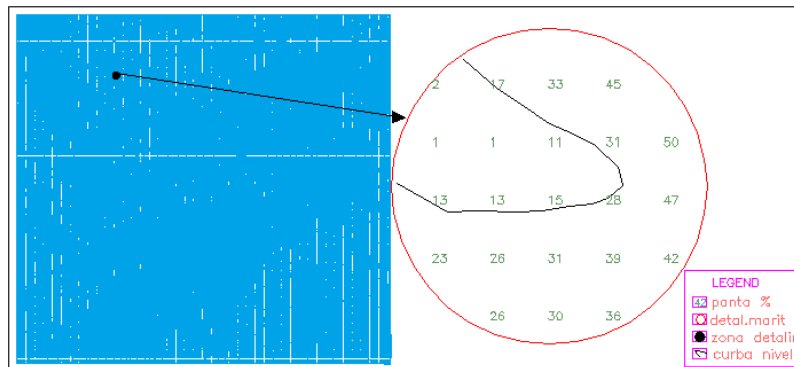


Figure 3.1. The map with slope data [1]

3.2. The thematic map of the slope dimension intervals

The thematic map of the studied area, executed on slope dimension intervals, presents the slope types distribution on the terrain in case. The design of the map implied data query from "MPANTA.TXT" file on six slope dimension intervals. (figure 3.2)

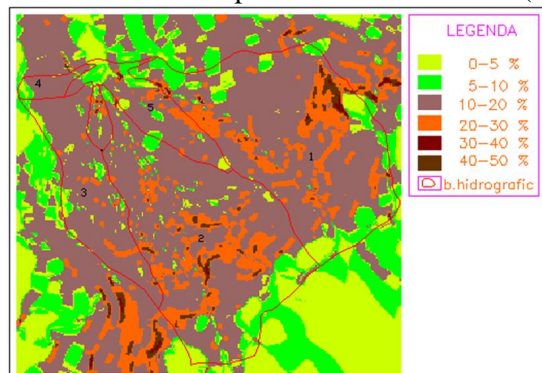


Figure 3.2. Thematic map of the study area picturing slope dimension intervals [1]

4. Conclusions

- Autodesk MAP 3D is an open platform that enables the development of any GIS application specific to several domains;
- The procedure for determining the slope can be used on any area of study numerically modeled;
- The application can be integrated into a larger one- such as rain/drain- in which the results obtained during the interpolation process become input data for the modeling process

5. References

1. Bofu C., 1999, *Contributions to the rating of optimal hydro-meteorological parameters needed in sizing hydraulic construction*, Ph. D. Diss. Techn. Univ. " Jassi,
2. Bofu C., Giurma R.C., 2000, *Numerical Model of Tinoasa Catchement Area*, Bul. Inst. Polit. Iasi, XLVI(L),1-4,s. Hidrot.
3. *Help-ul platformei Atodesk MAP 2014*