Zbigniew Piasek

Numerical approximation of large industrial excavations

Part two: Analysis of Software

Abstract

The second part of the paper concerns numerical approximation of analysed surfaces by three algorithms: CALCPAR, CALCABC, and CALCH. Calculations were carried out in FORTRAN language. The suggested calculation provides sufficient precision for mathematical modelling of the large industrial excavations.

1 Introduction

The considered surfaces are divided into “calculation zones”. Descriptions of zones division and generation of triangles net are presented in [10].

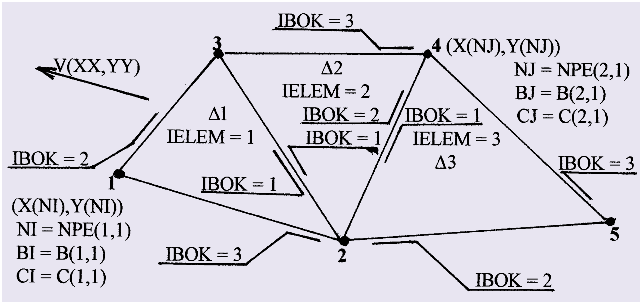
System calculations of the mathematical approximation of large industrial excavation surfaces are formulated in three algorithms: CALCPAR, CALCABC, and CALCH.

Algorithm CALCPAR involves calculation of curvature parameters. The second algorithm CALCABC computes the polynomial coefficients AT, BT, CT, and double surface area DELTA for all generated triangles from given “calculation zone”. The third analysed algorithm CALCH calculates the elevations of checking points, accordance with presented theory in part one of this paper.

At the stage of automatic generation [10] the codes KB have been ascribed to the sides of triangles. If there is KB = 1 (comparable with table on fig. 1), for the sides located inside “calculation zone”, curvature parameters are handed over from considered triangle to adjacent one. In such a case, parameters are calculated for the elevation function of higher degree. Parameters are notated as P_i^m and P_j^n or PAR.

On the edge of “calculation zone” the codes KB of sides in generated triangles were assumed equal to 2. If the KB = 2 (fig. 1), then curvature parameters of the sides are equal to zero and it means parameters PAR = 0.

In the algorithm CALCPAR for sides with codes KB = 1 the curvature parameters of:
– current triangle IELEM,
– its side IBOK, and
– current adjacent triangle IEL2
are calculated at the same time.



NPE

Number of triangle	Number of points in triangle		
1	1	2	3
	NPE(1,1)=1	NPE(1,2)=2	NPE(1,3)=3
2	4	3	2
	NPE(2,1)=4	NPE(2,2)=3	NPE(2,3)=2
3	5	4	2
	NPE(3,1)=5	NPE(3,2)=4	NPE(3,3)=2

KB

Number of triangle	Number of sides in triangle		
	IBOK = 1	IBOK = 2	IBOK = 3
1	1	2	2
2	1	1	2
3	1	2	2

NEL

Number of triangle	Number of sides in triangle		
	IBOK = 1	IBOK = 2	IBOK = 3
1	Δ2	0	0
2	Δ1	Δ3	0
3	Δ2	0	0

Fig. 1: Matrices of numbering the points in triangle (NPE), code sides (KB), and numbering of adjacent triangles (NEL).

2 Analysis of software

Calculations of curvature parameters
Algorithm CALCPAR

Input parameters:
X,Y,H – spacial co-ordinates from generated triangles net,
NPE – matrices numbering of points in generated triangles,
NELEM – quantity of generated triangles,
KB – matrices codes of sides in generated triangles,

NEL – matrices numbering of adjacent generated triangles.

5 | 6 | 7

```
      PROGRAM CALCPAR (X,Y,H,NPE,NELEM,
      KB,NEL,A,B,C,DELT,PAR)
      REAL X(500,3),Y(500,3),H(500,3),XT(3),YT((3),
      *AT(3),BT(3)
      REAL CT(3),A(500,3),B(500,3),C(500,3),PAR
      *(500,3),DELT(500)
      INTEGER NPE(500,3),KB(500,3),NEL(500,3),
      *NP(3)
      DO 3 IELEM = 1,NELEM
      DO 1 IBOK = 1,3
      XT(IBOK) = X(NPE(IELEM,IBOK))
1     YT(IBOK) = Y(NPE(IELEM,IBOK))
      CALL CALCABC (XT,YT,AT,BT,CT,DELTA)
      DO 2 IBOK = 1,3
      A(IELEM,IBOK) = AT(IBOK)
      B(IELEM,IBOK) = BT(IBOK)
2     C(IELEM,IBOK) = CT(IBOK)
3     DELT(IELEM) = DELTA
```

In the algorithm CALCPAR loop DO 3 runs successively by quantity of triangles NELEM, which has been generated. Internal loop DO 1 runs by numbers of sides in triangles. For example, in assumed triangle number IELEM = 1 we obtain instructions:

```
XT(1) = X(NPE(1,1))   ;   YT(1) = Y(NPE(1,1))
XT(2) = X(NPE(1,2))   ;   YT(2) = Y(NPE(1,2))
XT(3) = X(NPE(1,3))   ;   YT(3) = Y(NPE(1,3))
```

numbers of sides in triangle — numbers of sides — number of triangle

Next subroutine CALCABC calculates coefficients AT,BT,CT, and double surface area DELTA of a given triangle. All calculated variables are substituted to global matrices A,B,C, and DELT.

```
      DO 7 IELEM = 1,NELEM
      DO 7 IBOK = 1,3
      IF(KB(IELEM,IBOK).EQ.0) GO TO 7
      IF(KB(IELEM,IBOK).EQ.1) GO TO 4
      IF(ABS(KB(IELEM,IBOK)).EQ.2) PAR(IELEM,
      IBOK) = 0
      GO TO 7
4     IEL2 = NEL(IELEM,IBOK)
      DO 5 I = 1,3
5     IF(NEL(IEL2,I).EQ.IELEM) IBOK2 = I
      NI = NPE(IELEM,IBOK)
      NJ = NPE(IEL2,IBOK2)
      BI = B(IELEM,IBOK)
      CI = C(IELEM,IBOK)
      BJ = B(IEL2,IBOK2)
      CJ = C(IEL2,IBOK2)
      HJI = H(NJ) - H(NI)
      XX = X(NI) - X(NJ)
      YY = Y(NI) - Y(NJ)
      ODLIJ = SQRT(XX**2 + YY**2)
      XX = XX / ODLIJ
      YY = YY / ODLIJ
      SIX = 0
      SIY = 0
      SJX = 0
      SJY = 0
      DO 6 I = 1,3
      SIX = SIX + B(IELEM,I) * H(NPE(IELEM,I))
      SIY = SIY + C(IELEM,I) * H(NPE(IELEM,I))
      SJX = SJX + B(IEL2,I) * H(NPE(IEL2,I))
      SJY = SJY + C(IEL2,I) * H(NPE(IEL2,I))
6     CONTINUE
      SIX = SIX * XX
      SIY = SIY * YY
      SJX = SJX * XX
      SJY = SJY * YY
      SM = (SIX + SIY) / DELT (IELEM)
      SN = (SJX + SJY) / DELT(IEL2)
      BCI = (BI * XX + CI * YY) / DELT(IELEM)
      BCJ = (BJ * XX + CJ *YY) / DELT(IEL2)
      D = BCI / BCJ
      E = SM / BCJ
      F = -SN / BCJ
      PAR(IELEM,IBOK) = - D * (E + F) / (1 + D * D)
      PAR(IEL2,IBOK2) = PAR(IELEM,IBOK) * D +
      *E + F
      KB(IELEM,IBOK) = 0
      KB(IEL2,IBOK2) = 0
7     CONTINUE
      RETURN
      END
```

As before, similarly the loops IELEM = 1, NELEM and IBOK = 1,3 run successively by quantity of generated triangles and sides. Loop DO 7 calculates parameters of curvature on the base of described theory in the first part of this paper.

Three logical instructions check coding of:

- sides which lie opposite to considering apexes (IF(KB(IELEM,IBOK).EQ.0)),
- common side of two triangles (IF(KB(IELEM, IBOK).EQ.1)),
- sides on the edge of a given zone (IF(ABS(KB(IELEM,IBOK)).EQ.2)).

Example of coding system KB is presented in table 1.

Next, algorithm CALCPAR determines directional vector on the base of two points from two adjacent triangles. Then directional vector is normalising.

Co-ordinates of unit vector have the following form:

$$V_{unit}\left(\frac{V_x}{\sqrt{V_x^2+V_y^2}},\frac{V_y}{\sqrt{V_x^2+V_y^2}}\right)$$

In the first part of this paper, parameters of elevation function have the following form:

$$P_i^m = \frac{T_i^m(S_j^n - S_i^m)}{(T_j^n)^2 + (T_i^m)^2} \tag{1}$$

$$P_j^n = \frac{S_j^n - S_i^m}{T_j^n}\left(\frac{(T_i^m)^2}{(T_j^n)^2 + (T_i^m)^2} - 1\right) \tag{2}$$

In order to simplify the calculations we shall transform formula (1) and (2).

After transformation of formula (1) we obtain:

$$P_i^m = \frac{-\frac{T_i^m}{T_j^n}\left(\frac{S_i^m}{T_j^n} + \frac{S_j^n}{T_j^n}\right)}{1 + \left(\frac{T_i^m}{T_j^n}\ \frac{T_i^m}{T_j^n}\right)}$$

We assume:

$$D = \frac{T_i^m}{T_j^n} \ ; \ E = \frac{S_i^m}{T_j^n} \ ; \ F = \frac{S_j^n}{T_j^n}$$

The final form of formula (1) in subroutine CALCPAR is the following:

$$PAR\,(IELEM\ ,IBOK\,) = \frac{-D(E+F)}{1 + D * D}$$

In the first part of this paper following formulas have been used:

$$T_i^m = \frac{b_i^m}{\Delta^m} V_x + \frac{c_i^m}{\Delta^m} V_y \qquad (3)$$

$$T_j^n = \frac{b_j^n}{\Delta^n} V_x + \frac{c_j^n}{\Delta^n} V_y \qquad (4)$$

We assume the following notations:

$BCI = T_i^m$; $BI = b_i^m$; $CI = c_i^m$; $DELT\ (IELEM\ ) = \Delta^m$
$BCJ = T_j^n$; $BJ = b_j^n$; $CJ = c_j^n$; $DELT\ (IEL\,2) = \Delta^n$
$XX = V_x$; $YY = V_y$

The final forms of formulas (3) and (4) in considered subroutine are as follows:

$$BCI = \frac{BI * XX + CI * YY}{DELT\ (IELEM\ )} \ ;$$

$$BCJ = \frac{BJ * XX + CJ * YY}{DELT\ (IEL\,2)}$$

In the first part of this paper the following formulas have been considered:

$$S_i^m = \frac{1}{\Delta^m} \sum_{i=1}^{i=3} z_i^m (b_i^m V_x + c_i^m V_y) \qquad (5)$$

$$S_j^n = \frac{1}{\Delta^n} \sum_{j=1}^{j=3} z_j^n (b_j^n V_x + c_j^n V_y) \qquad (6)$$

Taking into account notations in analysed subroutine we obtain:

$$SM = S_i^m \ ; \ SN = S_j^n$$

$$H(NPE\,(IELEM\ ,I) = Z_i^m \ ; \ H(NPE\,(IEL\,2,I) = Z_j^n$$

$$B(IELEM\ ,I) = b_i^m \ ; \ B(IEL\,2,I) = b_j^n$$

$$C(IELEM\ ,I) = c_i^m \ ; \ C(IEL\,2,I) = c_j^n$$

From loop DO 6 I = 1,3 we obtain:

$$SIX = SIX + B(IELEM\ ,I) * H(NPE\,(IELEM\ ,I)) = \sum_{i=1}^{i=3} b_i^m z_i^m$$

$$SIY = SIY = C(IELEM\ ,I) * H(NPE\,(IELEM\ ,I)) = \sum_{i=1}^{i=3} c_i^m z_i^m$$

$$SJX = SJX + B(IEL\,2,I) * H(NPE\,(IEL\,2,I)) = \sum_{i=1}^{i=3} b_j^n z_j^n$$

$$SJY = SJY + C(IEL\,2,I) * H(NPE\,(IEL\,2,I)) = \sum_{i=1}^{i=3} c_j^n z_j^n$$

The final form of formulas (5) and (6) in algorithm CALCPAR are as follow:

$$SM = \frac{SIX + SIY}{DELT\ (IEL\,2)}$$

$$SN = \frac{SJX + SIY}{DELT\ (IEL\,2)}$$

The final form of formula (2) in algorithm CALCPAR is following:

PAR(IEL2,IBOK2) = PAR(IELEM,IBOK) * D +E + F

Before calculation of parameters PAR for successively sides of triangles or for next triangles IELEM, we check by means of instructions KB(IELEM,IBOK) and KB(IEL2,IBOK2) if code of sides are equal to zero. By this way we prevent calculations of parameters PAR once more.

Output parameters from algorithm CALCPAR:
A,B,C – coefficients of linear polynomial from considered triangles,
DELT – double surface area of considered triangles,
PAR – parameters of curvature.

Calculation of coefficients A,B,C, and parameter DELTA in analysed triangle. Algorithm CALCABC

Input parameters:
XT,YT – co-ordinates of points of triangles, which have been generated,

5 | 6 | 7

```
SUBROUTINE CALCABC (XT,YT,AT,BT,CT,
DELTA)
REAL XT(3),YT(3),AT(3),BT(3),CT(3)
X1 = XT(1)
X2 = XT(2)
X3 = XT(3)
Y1 = YT(1)
Y2 = YT(2)
Y3 = YT(3)
AT(1) = X2 * Y3 - X3 * Y2
BT(1) = Y2 - Y3
CT(1) = X3 - X2
AT(2) = X3 *Y1 - X1 * Y3
BT(2) = Y3 - Y1
CT(2) = X1 - X3
AT(3) = X1 * Y2 - X2 * Y1
BT(3) = Y1 - Y2
CT(3) = X2 - X1
DELTA = X2*Y3+X1*Y2+X3*Y1-X2*Y1-X3*
*Y2-X1*Y3
RETURN
END
```

In order to calculate coefficients AT,BT,CT, we have to solve the set of equations from theory presented in the first part of this paper. For example, assumed set of equations at apex 1 of triangle have the following form:

$$a_1 + b_1 x_1 + c_1 y_1 = 1$$
$$a_1 + b_1 x_2 + c_1 y_2 = 0$$
$$a_1 + b_1 x_3 + c_1 y_3 = 0$$

From this set of equations we obtain:

$$a_1 = x_2 y_3 - x_3 y_2 \quad ; \quad b_1 = y_2 - y_3 \quad ; \quad c_1 = x_3 - x_2$$
$$W = x_2 y_3 + x_1 y_2 + x_3 y_1 - x_2 y_1 - x_3 y_2 - x_1 y_3$$

"W" is equal to parameter DELTA. Similarly from the set of equations at apexes 2 and 3 we shall calculate coefficients: a_2,b_2,c_2 and a_3,b_3,c_3.

Output parameters from subroutine CALCABC:
AT,BT,CT – coefficients of linear polynomial from analysed triangles,
DELTA – double surface area of analysed triangles.

Calculations of the heights of the points in the generated triangles notated as IEL. Algorithm CALCH

Input parameters:
X,Y,H – spacial co-ordinates in apexes of triangle which has been generated.
A,B,C – coefficients of linear polynomial,
NPE – matrice of points in triangle,
PAR – parameters of curvatures,
DELT – double surface area of considered triangle,
IEL – number of considered triangle.

In algorithm CALCH are use notations L1,L2,L3, which overlap with notations from the first part of this paper and specifically concern the following set of equations:

$$L_1 = (a_1 + b_1 x + c_1 y) / 2\Delta$$
$$L_2 = (a_2 + b_2 x + c_2 y) / 2\Delta$$
$$L_3 = (a_3 + b_3 x + c_3 y) / 2\Delta$$

Into variables W1,W2,W3 are have substituted heights from apexes of a given triangle with number IEL, and into variables P1,P2,P3 the parameters of curvatures for sides of triangle are substituted.

5 | 6 | 7

```
PROGRAM CALCH (X,Y,H,A,B,C,NPE,PAR,
DELT,IEL,W)
REAL H(500),A(500,3),B(500,3),C(500,3),DELT
(500),PAR(500,3)
INTEGER NPE(500,3)
REAL L1,L2,L3
L1 = (A(IEL,1)+B(IEL,1)*X+C(IEL,1)*Y)/
*DELT(IEL)
L2 = (A(IEL,2)+B(IEL,2)*X+C(IEL,2)*Y)/
*DELT(IEL)
L3 = (A(IEL,3)+B(IEL,3)*X+C(IEL,3)*Y)/
*DELT(IEL)
W1 = H(NPE(IEL,1))
W2 = H(NPE(IEL,2))
W3 = H(NPE(IEL,3))
```

Table 1: Heights of measured points and values of relative errors

Number of Point	HR (Metre)	HL (Metre)	HN (Metre)	δ_{LN} (%)	δ_{NR} (%)
1	400.09	400.31	400.11	0.05	0.00
2	400.42	400.49	400.36	0.03	0.02
3	401.03	400.89	401.01	0.03	0.00
4	401.24	401.02	401.19	0.04	0.01
5	402.33	402.41	402.29	0.03	0.01
6	402.38	402.50	402.37	0.03	0.00
7	402.02	402.06	402.17	0.03	0.04
8	402.93	403.18	402.98	0.05	0.01
9	402.33	402.46	402.29	0.04	0.01
10	399.05	399.28	399.02	0.07	0.01
11	400.03	399.82	400.02	0.05	0.00
12	398.23	398.31	398.21	0.03	0.01
13	399.67	399.54	399.59	0.01	0.02
14	399.49	399.58	399.56	0.01	0.02
15	399.87	400.06	399.83	0.06	0.01
16	401.43	401.58	401.44	0.04	0.00
17	401.29	401.12	401.32	0.05	0.01
18	401.98	401.81	401.95	0.04	0.01
19	402.19	402.44	402.22	0.06	0.01
20	402.23	402.01	402.29	0.07	0.02
21	402.26	402.09	402.30	0.05	0.01
22	401.87	401.45	402.03	0.14	0.04
23	401.33	401.24	401.31	0.02	0.00
24	402.31	402.56	402.28	0.07	0.01
25	401.89	401.78	401.92	0.04	0.01
26	401.67	401.54	401.66	0.03	0.00
27	401.72	401.79	401.78	0.00	0.02
28	398.83	399.01	398.76	0.06	0.02
29	397.44	397.31	397.48	0.04	0.01
30	398.89	398.56	398.94	0.03	0.01
31	398.43	398.12	398.35	0.06	0.02
32	399.32	399.61	399.39	0.06	0.02
33	400.11	399.83	400.02	0.05	0.02
34	402.56	402.91	402.63	0.07	0.02
35	402.97	403.24	402.92	0.08	0.01
36	402.86	402.99	402.81	0.05	0.01
37	403.01	402.87	403.14	0.07	0.03
38	403.05	403.43	403.21	0.05	0.04
39	403.34	403.41	403.30	0.03	0.01
40	403.78	403.97	403.81	0.04	0.01

```
P1 = PAR(IEL,1)
P2 = PAR(IEL,2)
P3 = PAR(IEL,3)
W=L1*W1+L2*W2+L3*W3+P1*L1*(L2+L3)+
*P2*L2*(L1+L3)+P3*L3*(L1+L2)
RETURN
END
```

In the first part of this paper the equation of elevation function has been assumed in the following form:

$$z = L_1Z_1 + L_2Z_2 + L_3Z_3 + P_1L_1(L_2 + L_3) + P_2L_2(L_1 + L_3) + P_3L_3(L_1 + L_2)$$

It has related to calculation of height W in the given triangle.

Output parameter from algorithm CALCH:
W – heights of points in a given triangle.

3 Testing of proposed software and conclusions

The practical calculation based on the fragment of industrial excavation. Measurement on this area has been performed with the use electronic tacheometer TC – 2003. The obtained numbers of measured points are from 29 to 43 on the area of one hectare. On the considered area, additionally measurement has been performed, which involve 40 arbitrary measurement points. In consequence, the testing of presented in this paper software is based on 40 points. After generation of the net of triangles the mathematical approximation of measured zone has been performed by means of the elevation function of the first and of the second degree. The table 1 presents results of calculation in 40 measurement points.

In this table the following variables are introduced:
HR – analytically calculated heights,
HL – calculated heights by means of shape function of the first degree,
HN – calculated heights by means of shape function of the second degree,
δ – values of relative errors of measurement points,

Calculated relative errors δ_{NR} are related to heights HR, which are assumed to be accurate.

Then: δ_{NRmin} = 0.00 %, δ_{NRmax} = 0.04 %, and $\delta_{NRaverage}$ for 40 measurement points = 0.01 %. In consequence we can assume that calculated heights HN are accurate as well.

Table 1 shows that by application of shape function of the second degree we obtain the increase of approximation accuracy of analysed surface in relation to linear function. The range of increased accuracy is from $\delta_{LN\ min}$ = 0.0 % to δ_{LNmax} = 0.14 %, and $\delta_{LNaverage}$ for 40 measurement points = 0.05%.

The non-linear shape function adopted in this paper, which use parameters related to adjacent triangles allows for good excavation surface approximation and at the same time it gives a very effective calculation algorithms. Approximation of excavations can be performed with high accuracy. This precision is particularly useful for calculation of volume of excavated earth masses, if we consider costs of exploitation of large industrial excavations.

References

[1] AUTUME G., DE MASSON D.: L'interpolation par une regle flexible et ses applications en photogrammetrie numerique, XIII-th Congres of the ISP, presented paper, 1976.

[2] BAUHUBER F., ERLACHER V. and GUNTHER P.: Ein Programmsystem für die Behandlung digitaler Höhenmodelle, Geodätische Woche, Köln 1975.

[3] BEYER A.: Zur Erfassung flachen Geländes durch willkürlich verteilte Höhenpunkte, Vt 20, 1972.

[4] BOSMAN E. R., ECKHARD D. and KUBIK K.: Delft – a programme system for surface approximation, Bul.40, 1972.

[5] CONNELLY D. S.: An experiment in contour map smoothing on the ECU automated contouring system, CJ8, 1971.

[6] JUNKINS J. L., JANCAITIS J. R.: Mathematical terrain analysis, Proceedings of the 37 Th. Annual Meeting of the ACSM, 1971.

[7] PEUCKER T. K.: Computer Cartography, Resource Paper No 17, Washington D.C. 1972.

[8] PETRIE G.: Photogrammetric techniques of data equation for terrain modelling. Department of Geography and Topographic, University of Glasgow, 1987.

[9] PIASEK Z.: A numerical methods of approximation of given soil areas, Polisch Academy of Sciences, 1993.

[10] PIASEK Z.: Numerical method of cap silting Research of storage reservoirs. Copyright by Technical University of Technology, Kraków, 1998.

[11] PIASEK Z.: Determination of number of numerical integration steps on an industrial dump example. AVN, 8/9, 2000.

[12] RHIND D.: Automated contouring – an empirical evaluation of some differing techniques, CJ8, 1971.

[13] SCHUT G. H.: Review of interpolation methods for digital terrain models, XIII Congress of I.S.P., Helsinki, 1976.

[14] TORLEGARD K., OSTMAN A., LINDGREN R.: A comparative test of photogrammetrically sampled Digital Elevation Models. Royal Institute of Technology, S-10044, Stockholm, 1986.

[15] ZIENKIEWICZ O. C.: Finite elements method, Arkady, Warszawa 1972.

Address of the author:
Prof. ZBIGNIEW PIASEK D. Sc., Ph. D., M. Sc.
Cracow University of Technology
Division of Engineering Geodesy
31-155 Kraków
ul. Warszawska 24
Poland