Creating new user defined functions for 2D adjustment by parameter variation modelling

Ustvarjanje novih lastnih funkcij za modeliranje 2D posredne izravnave

MILIVOJ VULIĆ1, DUŠAN SETNIKAR2

1University of Ljubljana, Faculty of Natural sciences and Engineering, Department of Geotechnology and Mining Engineering, Aškerčeva cesta 12, SI-1000 Ljubljana, Slovenia; E-mail: milivoj.vulic@ntf.uni-lj.si
2Geodetski zavod Celje d.o.o., Ulica XIV. divizije 10, 3000 Celje, Slovenia; E-mail: dusan.setnikar@gz-ce.si

Received: September 25, 2007 Accepted: December 9, 2007

Abstract: Trigonometric network adjustments are executable with a wide range of programmes. For the modelling of a 2D adjustment by parameter variation[5], Excel was chosen due to its widespread use, accessibility and generally well-known basic use, furthermore also because of the easy scanning, flexibility in procedure determination and UDF support, the use of which adds considerably to the ease of scanning. The user defined functions arranged for an Excel environment that were used in the referred adjustment[1][5] are presented. Each UDF presentation consists of an overview of terms, directions of use and the simple uniform case. Created UDFs can be downloaded[3][4] for free as Add-ins for an Excel environment. A new approach, referred to as the “switching off/on of data of single measurements or a group of measurements of the same sort,” is also presented. The adjustment of a simple imaginary trigonometric network consisting of five measurement points is also included in the article. As an addition[1] to the latter, a visual review of UDF use, filmed with the programme CamStudio[2] - which enables a beginner to learn how to use these functions - was included.

Izvleček: Izravnavo trigonometrične mreže lahko izvedemo z veliko programi. Za izdelavo modela posredne 2D izravnave[1] je bil izbran Excel, ker je splošno dostopen, osnove uporabe so znane, je pregleden in prilagodljiv pri izvajanju procedure, hkrati pa podpira tudi uporabo lastnih funkcij. Uporaba lastnih funkcij doprinese predvsem k večji preglednosti. V članku so predstavljene funkcije, ki so bile prilagojene za Excelovo okolje in so bile uporabljene pri referenčni izravnavi[1][5]. Vsaka lastna funkcija je predstavljena z opisom uporabljenih pojmov, navodilom za uporabo in enotnim preprostim primerom. Vse te funkcije so dostopne[3][4] zastonj v obliki dodat-

**Key words:** adjustment by parameter variation, UDF, MS Excel, switching off/on measurements

**Ključne besede:** posredna izravnava, UDF, MS Excel, izklop/vklop meritev

**INTRODUCTION**

When adjusting a trigonometric network, we process a substantial amount of data, which is why the use of any computer programme that performs these calculations without any substantial errors is a reasonable step to take. We can perform a selection from a wide range of specialised programmes (TRIM, GEOS…) as well as programmes that enable the user to adjust an individual procedure to its demands (such as Matlab, Mathematica, Scilab, Maple, Excel, OooCalc…).

Specialised programmes are more user-friendly, but also rather expensive. Besides that, these programmes function on the basis of the “black box,” in which data is entered, and from this, the results are returned. However, in such a case, we have no insight into the actual performance and thus cannot check the accuracy other than by a reference calculation. Due to the fact that we know their true function, with proper programming tools, our own customised application can be created. Each programming tool has both advantages and disadvantages. The main factors affecting what our chosen programme is to be are the malleability of the working environment to our demands and calculation process auditing, sometimes even the data processing speed. Programmes such as Matlab, Mathematica - to name a couple of examples - are usually a lot faster; however, they are far less auditable, more expensive and, due to much needed specific previous knowledge, applicable only to a few users. On the other hand, we have programmes such as MS Excel and OOo.Calc, available for a low price or for free, respectively. Their basic use is well-known, they are flexible in procedure determination and can easily be scanned. Specialised programmes are meant to be used in fluent projects by poorly educated users. They are intended for users who adopt results as optimal or accurate enough for their needs, remising the presence of eventual larger errors, which could be annulled or reduced to an acceptable range through the use of a proper approach. Individually adjusted programmes are more research-oriented and therefore intended for users wanting to know the influences on calculation accuracy, there due to the acquiring of satisfactory results through the rejection of bad measurements or with simulated measurements, creating a model convenient for the task set. The re-
Creating new user defined functions for 2D adjustment by parameter variation[1] and the simplified case presented as an attachment to this article offer us this option.

Due to all the stated reasons, and because MS Excel supports creating user defined functions with the help of MS Visual Basic for Applications, MS Excel was chosen to form a model of a 2D adjustment by parameter variation[1]. This is a compromised solution, using both an individual procedure and user defined functions, which are actually specialised sub-programmes for defined calculative operations; the kind that are again part of an adjustment procedure as a whole. UDFs are still small black boxes, but their algorithms are presented further in the continuation of this article and a visual presentation of UDF use is available[1], thus making this model acceptable for a lower level of theoretical knowledge as well. In turn, the model enables the calculation of an extensive trigonometric network:
- in several epochs,
- on optional locations,
- merely by entering field measurements.

Created UDFs can be downloaded[3],[4] for free as Add-ins for Excel environment. Since they can be optionally complemented, the authors would appreciate any forwarded comments, experiences or eventual malfunctions in their use.

Due to a limited printing space, the case enclosed to this article is a hypothetical small-scale application of a referred adjustment model. It consists of five measurement points, four of them known (the visures) and one unknown (the station). The coordinates of the unknown point are calculated when three known points (T1, T2 and T3) are used. The calculation is repeated when a new known point (new*) is introduced. The value of \( m_0 \) is a reference to evaluate the benefits of introducing a new known point into the adjustment process. Its decreased value (see the enclosed case in Table 15) clearly shows that the accuracy of coordinate determination of the unknown point has been improved. Analogically, we can also conclude that the

Table 1. The imaginary data set for the network adjustment
Tabela 1. Namišljeni podatki za izravnavo mreže (dolžina, smer, stojišče, vizurne točke, meritev, natančnost, utež)
direction measurement to point new* is more accurate than the distance measurement to the same point.

The enclosed printed case cannot express the dynamic nature of the referred adjustment and moreover doesn’t include all the UDFs presented here. For these reasons, the reader is encouraged to visit the NTF site and download the large-scale adjustment model including all of the UDFs presented in the article at hand.

**WHAT IS A UDF?**

A user defined function (referred to simply as a “UDF” further on in the article) functions as an add-in in the MS Excel programme tools platform. It is virtually a part of MS Excel and is simply summoned from the function line. This additional function enables faster work and adds to table transparency. An optional number of UDFs can be added, although it is recommended to add only those used in the specific task in order to achieve a higher processing speed.

UDFs are similar to macros, but with a less complex code. Their benefits are:
- creating a complex or custom math function,
- simplifying formulas that would otherwise be extremely long “mega formulas”,
- custom text manipulation,
- advanced array formulas and matrix functions,
- a UDF’s programme code can be locked, preventing its unauthorised alteration,
- an add-in is available without the need to open new worksheet.

**ADJUSTMENT BY PARAMETER VARIATION WITH THE USE OF A UDF**

The basis for the calculations is presented by the adjustment theory. The setting out method is used and distances and directions are measured. Field measurements or their simulations are used in UDFs in order to acquire matrices of equation coefficients; to eliminate $z$ through Gaussian elimination in order to get residuals equations.

The adjustment consists of two separate parts. First we acquire the design matrix of the equation coefficients by assuming the sought-for coordinates. In the second part, we use the field measurements and perform a Gaussian elimination. The matrix of normal equation coefficients, as well as the matrix of unknowns, is formed. The result is the residuals equation:
Creating new user defined functions for 2D adjustment by parameter...

\[ v = Ax + f \]  

where \( v \) is the vector of the residuals, \( A \) is the design matrix of observation equations, \( x \) is a vector of unknowns and \( f \) is a vector of absolute terms.

The weighed least square method is used. When the product of \( v^TPv \) (\( v \) is a vector of residuals, \( P \) a matrix of weights, acquired in the base of expected measurements accuracy) is the smallest, the result is optimal for the data entered. A vector of unknown parameters is then added to the assumed coordinate data, thus iterating its value until we reach satisfactory results, but of course always according to the accuracy of our field measurements.

**HOW TO USE A UDF?**

Furthermore, the way we use our own - personally written - functions is presented. We have carried out the adjustment with the use of following functions\[^{[6]}\].

**dLine**

This function returns the coefficient values for the direction station_visure, thus giving us its angle towards the oriented direction, \( \Delta z \) and normalizing all the coefficients.

**Syntax:** \( \text{dLine} \) (yStation;xStation;yVisure; xVisure;grade;minute;second)

**Legend:**
- \( y_{\text{Station}} \) - [m] is the \( y \) coordinate of the station
- \( x_{\text{Station}} \) - [m] is the \( x \) coordinate of the station
- \( y_{\text{Visure}} \) - [m] is the \( y \) coordinate of the visure
- \( x_{\text{Visure}} \) - [m] is the \( x \) coordinate of the visure
- \( \text{grade} \) - [°] is the angle between the oriented direction and the visured direction, rounded off to a full number
- \( \text{minute} \) - [‘] is the hexadecimal part of the angle between the oriented direction and the visured direction, rounded off to a full number
- \( \text{second} \) - ["] is the hexadecimal part of the minute (see above), rounded off to full a number or to one decimal place

**Use:**
1. We select a field of size 1×5 (1 row for the direction, 1 column for \( \Delta z \), 2 columns for the station coefficients, 2 columns for the visure coefficients).
2. From the function line, we select UDF, then \( \text{dLine} \).
3. In the fields opened, we enter the data listed in the legend. Before closing the window, we confirm the data entered by pressing Ctrl+Shift+Enter.

**dLineW**

This function returns the matrix of coefficients of the values for all directions of station_visure, thus giving us their angles towards the oriented directions, \( \Delta z \) and normalizing all the coefficients. All the coefficients for the visures are listed in two columns (for \( y \) and \( x \)).

**Syntax:** \( \text{dLine} \) (Station;Visure;to25;grade;minute;second)

**Legend:**
- \( \text{Station} \) - the name of the station, taken from to25
- \( \text{Visure} \) - the name of the visure, taken from to25
- \( \text{to25} \) - is the standard geodetical formulary.
Table 2. The initial and calculated points coordinates and mean square error for T point coordinates

Tabela 2. Začetne in izračunane koordinate točk in srednji kvadratni pogrešek določitve koordinat točke T

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>points/coordinates</td>
<td>[m]</td>
<td>[m]</td>
<td>[m]</td>
<td>[m]</td>
<td>[m]</td>
</tr>
<tr>
<td>14</td>
<td>MT_1</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>MT_2</td>
<td>800</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>MT_3</td>
<td>1000</td>
<td>400</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>T</td>
<td>500</td>
<td>400</td>
<td>0.0055</td>
<td>0.0013</td>
<td>500.0055</td>
</tr>
<tr>
<td>18</td>
<td>new*</td>
<td>1000</td>
<td>900</td>
<td>0</td>
<td>0</td>
<td>1000</td>
</tr>
</tbody>
</table>

\[ X^2 = 3.0 \times 10^{-5} \]

Table 3. Calculated coefficient values for directions station_visure and vice versa

Tabela 3. Izračunani smerni koeficienti za smeri stojišče_vizurna točka in obratno

\[ \begin{array}{cccc}
L & M & N & O \\
1 & z & \text{station/visure (y)} & \text{station/visure (x)} \\
-1 & 0 & 0.002 & 0 \\
\end{array} \]

\[ \text{for} \text{ (dLine(B17,C17,B16,C16,D5,E5,F5))} \]

\[ \text{direction coefficients} \]

Table 4. Calculated matrix of normalized coefficient values for directions station_visure and vice versa and as well for a vector of absolute terms

Tabela 4. Izračun matrike normaliziranih vrednosti smernih koeficientov za smeri stojišče_vizurna točka in obratno ter za vektor popravkov \( f \)

\[ \text{for} \text{ (dLineW(A$17,A16,A14,C18,D5,E5,F5)M12:X13))} \]

\[ \text{direction coefficients} \]

Table 5. Calculated matrix of normalized value for \( \Delta z \) and coefficient values for all directions station_visure and vice versa, for a vector of absolute terms

Tabela 5. Izračun matrike normaliziranih vrednosti \( \Delta z \) in vseh smernih koeficientov za smeri stojišče_vizurna točka in obratno, za vektor popravkov \( f \)

\[ \text{for} \text{ (dLineW(B3,A14,C18,D5,E5,F5))} \]

\[ \text{for} \text{ (dLineW(A$17,A16,A14,C18,D5,E5,F5))} \]

\[ \text{direction coefficients} \]

\[ \begin{array}{ccccccccccc}
L & M & N & O & P & Q & R & S & T & U & V & W & X \\
1 & z & \text{MT_1} & \text{MT_1} & \text{MT_2} & \text{MT_2} & \text{MT_3} & \text{MT_3} & T & T & \text{new*} & \text{new*} & \text{new*} \\
-1 & 0 & 0 & 0 & 0 & 0 & -0.002 & -0 & 0.002 & 0 & 0 & 839270 & 0 \\
\end{array} \]

RMZ-M&G 2007, 54
That is the table of coordinates assigned to all points in a trigonometric network, with which we operate in this function. It consists of one column for point names and two columns for the $y$ and $x$ values of the point coordinates. Where the coordinates are not known, we simply assume their values. Use of to25 in dLineW returns the table of coefficient values assigned to each direction_visure grade, minute, second (see previous function).

**Use:**
1. We select a field of dimensions $1 \times 6$ (1 row for the direction, 1 column for $\Delta z$, 2 columns for the station coefficients, 2 columns for the visure coefficients, 1 column for the vector of absolute terms).
2. From the function line, we select UDF, then dLineW.
3. In the fields opened, we enter the data listed in the legend. Before closing the window, we confirm the data entered by pressing Ctrl+Shift+Enter.

**dLineWall**
This function returns a matrix of coefficient values for all directions of station_visure, thus giving us their angles towards the oriented directions, $\Delta z$ as well as normalizing all the coefficients and the vector of residuals. Each coefficient for the station or the visures (the number of visures is $m$) is showed in a separate column and row. One column is reserved for vectors of absolute terms only.

**Syntax:** dLine (yStation;xStation;yVisure;xVisure;to25;grade;minute;secund;list)

**Legend:**
$y_{Station}, x_{Station}, y_{Visure}, x_{Visure}, grade,$
$minute, second; to25$ (see previous function)

**listC** - is the table of measurement points and their appropriate coordinate names ($y$ and $x$)

**Use:**
1. We select a field of size $n \times (2m+4)$ ($n$ rows for $n$ directions, 1 column for $\Delta z$, 2 columns for the station coefficients, $2m$ columns for the visure coefficients, 1 column for the residuals).
2. From the function line, we select UDF, then dLineWall.
3. In the fields opened, we enter the data listed in the legend. Before closing the window, we confirm this data entered by pressing Ctrl+Shift+Enter.

**dLineWallZ**
This function performs the Gaussian elimination of $\Delta z$ and returns the matrix of coefficient values for all directions of station_visure, thus giving us their angles towards their oriented directions; $\Delta z$ is annulled. Each coefficient for the station and all visures is shown in a separate column and row. One column is reserved for the vector of residuals alone.

**Syntax:** dLine (inputKernel;to25;distant;listC)

**Legend:**
to25, listC (see previous function)

**inputKernel** - is a table $5 \times n$ of the named station and visure points and their belonging measured directions (given in grades, minutes and seconds)

**distant** - is the distance between the station and visure point, reduced on a Gauss-Krüger projection

**Use:**
1. We select a field of size $n \times (2m+4)$ ($n$
Table 6. Calculated matrix of reduced coefficient values for all directions station_visure and vice versa and as well for a vector of absolute terms after the Gaussian elimination of $\Delta z$

<table>
<thead>
<tr>
<th></th>
<th>MT_1</th>
<th>MT_1</th>
<th>MT_2</th>
<th>MT_2</th>
<th>MT_3</th>
<th>MT_3</th>
<th>T</th>
<th>T</th>
<th>new*</th>
<th>new*</th>
</tr>
</thead>
<tbody>
<tr>
<td>z</td>
<td>1.8E-04</td>
<td>0.0012</td>
<td>0.0004</td>
<td>0.0003</td>
<td>0.0005</td>
<td>0.0008</td>
<td>-0.0023</td>
<td>-0.0003</td>
<td>0.0003</td>
<td>0.334237</td>
</tr>
<tr>
<td>0</td>
<td>0.0003</td>
<td>-0.0004</td>
<td>-0.0012</td>
<td>-0.0009</td>
<td>0.0005</td>
<td>0.0012</td>
<td>0.00055</td>
<td>-0.0003</td>
<td>-0.0003</td>
<td>-2.865763</td>
</tr>
<tr>
<td>0</td>
<td>0.0003</td>
<td>-0.0004</td>
<td>0.0004</td>
<td>0.0003</td>
<td>0.0002</td>
<td>-0.0005</td>
<td>0.00135</td>
<td>-0.0003</td>
<td>0.0003</td>
<td>1.965763</td>
</tr>
<tr>
<td>0</td>
<td>0.0003</td>
<td>-0.0004</td>
<td>0.0004</td>
<td>0.0003</td>
<td>0.0005</td>
<td>-0.0015</td>
<td>0.00035</td>
<td>0.0008</td>
<td>-8E-04</td>
<td>0.365763</td>
</tr>
</tbody>
</table>

Table 7. Calculated coefficients of a residual equation for a distance measurement

Table 8. Calculated coefficients of a residual equation for a distance measurement and vector of residuals

Table 9. Calculated coefficients of a residual equation for all distance measurements and vector of residuals
rows for \( n \) directions, 1 column for \( \Delta z \),
2 columns for the station coefficients,
2m columns for the visure coefficients,
1 column for absolute terms).

2. From the function line, we select UDF, then \texttt{dLineWallZ}.

3. In the fields opened, we enter the data listed in the legend. Before closing the window, we confirm the data entered by pressing Ctrl+Shift+Enter.

\textbf{dDist}

This function returns the coefficients of a residual equation for distance measurement.

\textbf{Syntax}: \texttt{dDist (yStation;xStation;yVisure; xVisure;distant)}

\textbf{Legend}:
\( y\text{Station}, x\text{Station}, y\text{Visure}, x\text{Visure},\text{distant} \) (see previous functions)

\textbf{Use}:
1. We select a field of size \( 1 \times 4 \) (1 row for the distance, 2 columns for the station coefficients, 2 columns for the visure coefficients).
2. From the function line, we select UDF, then \texttt{dDist}.
3. In the fields opened, we enter the data listed in the legend. Before closing the window, we confirm this data entered by pressing Ctrl+Shift+Enter.

\textbf{dDistW}

This function returns the coefficients of the residual equation for distance measurement and the vector of residuals. Each coefficient for the station and all the visures (the number of visures is \( m \)) are shown in a separate column.

\textbf{Syntax}: \texttt{dDistW (Station; Visure; to25; distant)}

\textbf{Legend}:
\( \text{station, visure, to25, distant, ListC} \) (see previous functions)

\textbf{Use}:
1. We select a field of size \( 1 \times (2m+3) \) (1 row for the direction, 2 columns for the station coefficients, 2\( m \) columns for the visure coefficients, 1 column for the vector of absolute terms).
2. From the function line, we select UDF, then \texttt{dDistWall}.
3. In the fields opened, we enter the data listed in the legend. Before closing the window, we confirm this data entered by pressing Ctrl+Shift+Enter.
4. We can expand this field in order to acquire the coefficients and vector of residuals also for the other distance measured in the same network. We
do this simply by selecting the field $1 \times (2m+3)$ and dragging it down, thus expanding it to as many rows as we have distances measured.

**takeDiagMatrix**

This function returns the values on a diagonal of a quadratic $n \times n$ matrix as a vector of $1 \times n$ dimensions.

**Syntax:** takeDiagMatrix(matrix)

**Legend:**

- **Matrix** - any quadratic matrix

**Use:**

1. We select a field of size $1 \times n$ ($n$ for the number of rows = the number of columns in a quadratic matrix).
2. From the function line, we select UDF, then **takeDiagMatrix**.
3. In the fields opened, we enter the data by selecting a quadratic matrix. Before closing the window, we confirm the data entered by pressing Ctrl+Shift+Enter.

**createListC**

This function returns a table of measurement points and their belonging coordinate names ($y$ and $x$), vectors of absolute terms and, when operating with direction measurements, also with $\Delta z$ (in this case, this is the name of a parameter and not a value).

**Syntax:** createListC(list)

**Legend:**

- **list** - the name of the coordinates participating in a trigonometric network; the list presents one column of to25 formulary

**Use:**

1. We select a field of size $2 \times (2p+1+1)$ ($p$ columns for the number of network points, 1 column for the vector of absolute terms).
2. From the function line, we select UDF, then **createListC**.
3. In the fields opened, we enter the list of point names (taken from to25 formulary). Before closing the window, we confirm the data entered by pressing Ctrl+Shift+Enter.

**EllipseW**

This function returns a field the contents of which are then imported by AutoCAD, resulting in the depiction of an ellipse.

**Syntax:** EllipseW(Ycenter, Xcenter, Zcenter, MajorSemi, MinorSemi, Rotation, Ratio)

**Legend:**

- **Ycenter** - the $Y$ coordinate of the ellipse
- **Xcenter** - the $X$ coordinate of the ellipse
- **Zcenter** - the $Z$ coordinate of the ellipse
- **MajorSemi** - the ellipse major semi-axis
- **MinorSemi** - the ellipse minor semi-axis
- **Rotation** - the azimuth of the major semi-axis
- **Ratio** - the scale at which the ellipse is drawn

**Use:**

1. Enter the values of the demanded data separately in 7 consecutive fields in one row.
2. Choose one field, open the **EllipseW** function line, enter the data demanded and press Enter.
3. Copy the same field, open an AutoCAD file and paste the data from the selected field. The ellipse gets drawn.

**pedaleW**

This function returns the values of coordinates for each pedale-forming point. The number of pedale-forming points is recip-
Table 10. Quadratic $n \times n$ matrix and calculated values on its diagonal as a vector of $I \times n$ dimensions (diagQxx)

Tabela 10. Kvadratna matrika $n \times n$ in izračunane vrednosti na njeni diagonali v obliki vektorja $I \times n$

Table 11. Table of measurement points and respective coordinate names, expanded with fields for vectors of absolute terms and $\Delta z$

Tabela 11. Tabela merskih točk in pripadajočih neznank, razširjena s polji za popravek $f$ in za orientacijsko smer $\Delta z$

Table 12. Calculation of values needed for drawing an ellipse in AutoCAD

Tabela 12. Izračun potrebnih koeficientov za izris elipse v AutoCAD-u (elementi kovariančne matrike neznank, parametri elipse in merilo izrisa)
rocal to AngleStepH.

**Syntax:** pedaleW(Ycenter,Xcenter,Zcenter,MajorSemi,MinorSemi,RotationH, AngleStepH,Ratio)

**Legend:**

*Ycenter, Xcenter, Zcenter, MajorSemi, MinorSemi, Ratio* (see previous function)

*RotationH* - the azimuth of the major semi-axis

*AngleStepH* - defines how wide an arc is, approximated with the line between two neighbouring pedale-forming points (e.g. the smaller the step, the higher the accuracy of the contour drawn)

**Use:**

1. Enter the values of the demanded data separately in 8 consecutive fields in one row.
2. Choose $3 \times n$ fields ($n \ldots 360$° divided by the angle step used, e.g. if you are drawing a circle and the angle step is 10, then you get $n=36$, thus getting the number of chords forming an approximate closed contour of a circle), open the pedaleW function line, enter the data demanded and press Enter. (P.S. When changing parameters afterwards, this can be done to all the parameters apart from the angle step, as the decrease in the angle step will cause a gap in the contour).
3. Choose $3 \times 1$ fields necessary for adding the coordinate data of the first point beginning the contour, thus forming a closed contour of the curve (this field must be filled manually).
4. Copy all $3 \times (n+1)$ fields, open an AutoCAD file and paste the data from the selected field. Only the pedale-forming points get drawn. Since it results in better visual conception, it would be reasonable to use the **lineW** function to draw the pedale contour subsequently.

**lineW**

This function returns the fields the contents of which are then imported by AutoCAD, where they result in the drawing of the contour of the pedale, approximated with the lines defined by this very function.

**Syntax:** pedaleW(Yfrom,Xfrom,Zfrom,Yto,Xto,Zto)

**Legend:**

*Yfrom* - the value for Y for the first point of the line

*Xfrom* - the value for X for the first point of the line

*Zfrom* - the value for Z for the first point of the line

*Yto* - the value for Y for the second point of the line

*Xto* - the value for X for the second point of the line

*Zto* - the value for Z for the second point of the line

**Use:**

1. Choose 1 field, open the LineW function line, enter the data demanded and press Enter.
2. Stretch this field to another $n$ consecutive fields in the column ($n \ldots$the same number as used in pedaleW).
3. Copy all the $(n+1)$ fields, open an AutoCAD file and paste the data from the selected field. The contour of the pedale gets drawn.
Table 13. Calculated values of coordinates for each pedale-forming point

Tabela 13. Izračun vrednosti koordinat, ki sestavljajo pedalo (elementi kovariančne matrike neznank, parametri elipse, merilo izrisa, točke pedale in kontur)

<table>
<thead>
<tr>
<th>L</th>
<th>M</th>
<th>N</th>
<th>O</th>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>S</th>
<th>T</th>
<th>U</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>mer. točka</td>
<td>elements of covariance matrix of unknowns</td>
<td>ellipse parameters</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$Q_{yy}$</td>
<td>$Q_{xy}$</td>
<td>$Q_{xx}$</td>
<td>$y_{cent}$</td>
<td>$x_{cent}$</td>
<td>$z_{cent}$</td>
<td>major semi $A$</td>
<td>minor semi $B$</td>
<td>$\theta$</td>
<td>ACAD</td>
</tr>
<tr>
<td>T</td>
<td>0.0001651</td>
<td>0.0000489</td>
<td>0.0000948</td>
<td>500</td>
<td>0.00055</td>
<td>500,00013</td>
<td>500</td>
<td>0.00239178</td>
<td>0.00239165</td>
<td>-54.2739</td>
</tr>
</tbody>
</table>

ellipse $e = 500.00554879576, 400.00127189378, 500.489.063861101369, 401.397868096414, 2.39164616676024$

$K = (Q_{yy} - 2Q_{xy})^2 + Q_{xx}^2$

$A^2_m = m^2(Q_{yy} + Q_{xx})$

$B^2_m = m^2(Q_{yy} - Q_{xx})$

Table 14. Calculated values of coordinates for each pedale contour-forming line

Tabela 14. Izračun vrednosti koordinat za linije, ki sestavljajo konturo pedale (točke pedale in linije kontur)

<table>
<thead>
<tr>
<th>L</th>
<th>M</th>
<th>N</th>
<th>O</th>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>S</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>pedele points</td>
<td>$z_{cent}$</td>
<td>contour</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U5030,10</td>
<td>400.0132304</td>
<td>500</td>
<td>line</td>
<td>500.00554879576, 400.0013230352175, 500.00554879576, 400.0013230352175, 500.00554879576, 400.0013230352175</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>500.011528</td>
<td>400.011628</td>
<td>500</td>
<td>line</td>
<td>500.00554879576, 400.0013230352175, 500.00554879576, 400.0013230352175</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

RMZ-M&G 2007, 54
**Switching the Measurement Data Off/on**

Through introducing a switcher (0 or 1; see fields 4F and 4I and subordinary fields B7 to B14 and L7 to L14 of the enclosed case in table 15), we gain the option to improve the final results by excluding bad single or group measurements from the input data set. It also gives us the option of defining the measurement points as either known or unknown. Once the specific data have been switched off/on, this affects all the other subordinate data in the adjustment. In combination with other calculative tools (i.e. the mean square error), we can evaluate the influence of the specific data on the adjustment results and reliability of the trigonometric network, be it either real or simulated.

**Conclusions**

When adjusting a trigonometric network, a substantial amount of data is treated. In order to achieve better programme performance and easier scanning, UDFs are introduced into Excel. On the basis of the data entered, a UDF can execute algorithms or treat non-numeric data. The optional exporting of UDF derivates to CAD programs is an additional benefit of the Excel environment, thus offering a graphical presentation of the calculated results. UDFs were used considerably in the adjustment model[1], thus saving time and space for the procedure. An additional approach used in this adjustment is switching bad measurements off/on in order to make the model dynamic and consecutively giving us the option to acquire more accurate results.

The case enclosed to this article is a printed small-scale version, and is unfit to reveal the dynamic nature of the referred adjustment, which is another reason why the reader is encouraged to visit the NTF site[1] and download the large-scale adjustment model, which includes all the UDFs presented here.

**Povzetek**

Pri izravnavi trigonometrične mreže obdelujemo večje količine podatkov. Excelove lastne funkcije (UDF) pripomorejo k večji učinkovitosti programa in omogočajo preglednost izračunov. Z lastnimi funkcijami obdelujemo vnešene numerične ali nenumerične podatke. Dodatna prednost UDF je možnost izvoza nekaterih rezultatov v CAD programe in s tem tudi njihova grafična predstavitev. UDF so bile v precejšnji meri uporabljane v referenčnem modelu izravnave[1], s čimer smo za obdelavo prihranili precej časa in prostora. Dodatno je predstavljen tudi nov pristop, ki omogoča izklop/vklop slabih meritev, kar nam omogoča pridobitev natančnejših rezultatov in hkrati naredi model bolj dinamičen.

Primer, priložen k temu članku, je zgolj ponostavljena verzija referenčnega modela in ne razkrija njegove dinamičnosti, kar je dodaten razlog za obisk NTF strani[1], od koder si lahko prenesete neokrnjeni model z vsemi predstavljenimi lastnimi funkcijami.
Creating new user defined functions for 2D adjustment by parameter...

<table>
<thead>
<tr>
<th>Table 15. Introductory overview of input elements and some control data for simulated case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tabela 15. Pregledni prikaz vhodnih elementov in nekaterih kontrolnih rezultatov za simulirani primer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>element</th>
<th>description</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td></td>
<td></td>
</tr>
<tr>
<td>K</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

RMZ-M&G 2007, 54
REFERENCES


