Spreadsheets in Education (eJSiE)

Volume 3 | Issue 1 Article 7

6-12-2009

3D Graphics with Spreadsheets

Jan Benacka
Constantine the Philosopher University, Nitra, Slovakia, jbenacka@ukf.sk

Follow this and additional works at: http://epublications.bond.edu.au/ejsie

This work is licensed under a Creative Commons Attribution-Noncommercial-No Derivative Works 4.0 License.

Recommended Citation
Benacka, Jan (2009) 3D Graphics with Spreadsheets, *Spreadsheets in Education (eJSiE)*: Vol. 3: Iss. 1, Article 7.
Available at: http://epublications.bond.edu.au/ejsie/vol3/iss1/7

This Regular Article is brought to you by the Bond Business School at ePublications@bond. It has been accepted for inclusion in Spreadsheets in Education (eJSiE) by an authorized administrator of ePublications@bond. For more information, please contact Bond University's Repository Coordinator.

3D Graphics with Spreadsheets

Abstract

In the article, the formulas for orthographic parallel projection of 3D bodies on computer screen are derived using secondary school vector algebra. The spreadsheet implementation is demonstrated in six applications that project bodies with increasing intricacy – a convex body (cube) with non-solved visibility, convex bodies (cube, chapel) with solved visibility, a coloured convex body (chapel) with solved visibility, and a coloured non-convex body (church) with solved visibility. The projections are revolvable in horizontal and vertical plane, and they are changeable in size. The examples show an unusual way of using spreadsheets as a 3D computer graphics tool. The applications can serve as a simple introduction to the general principles of computer graphics, to the graphics with spreadsheets, and as a tool for exercising stereoscopic vision. The presented approach is usable at visualising 3D scenes within some topics of secondary school curricula as solid geometry (angles and distances of lines and planes within simple bodies) or analytic geometry in space (angles and distances of lines and planes in E3), and even at university level within calculus at visualising graphs of z = f(x,y) functions. Examples are pictured.

Keywords

orthographic parallel projection, spreadsheets, 3D graphics

Distribution License

This work is licensed under a Creative Commons Attribution-Noncommercial-No Derivative Works 4.0 License.

This regular article is available in Spreadsheets in Education (eJSiE): http://epublications.bond.edu.au/ejsie/vol3/iss1/7

3D Graphics with Spreadsheets

Abstract

In the article, the formulas for orthographic parallel projection of 3D bodies on computer screen are derived using secondary school vector algebra. The spreadsheet implementation is demonstrated in six applications that project bodies with increasing intricacy – a convex body (cube) with non-solved visibility, convex bodies (cube, chapel) with solved visibility, a coloured convex body (chapel) with solved visibility, and a coloured non-convex body (church) with solved visibility. The projections are revolvable in horizontal and vertical plane, and they are changeable in size. The examples show an unusual way of using spreadsheets as a 3D computer graphics tool. The applications can serve as a simple introduction to the general principles of computer graphics, to the graphics with spreadsheets, and as a tool for exercising stereoscopic vision. The presented approach is usable at visualising 3D scenes within some topics of secondary school curricula as solid geometry (angles and distances of lines and planes within simple bodies) or analytic geometry in space (angles and distances of lines and planes in E_3), and even at university level within calculus at visualising graphs of $z = f(x,y)$ functions. Examples are pictured.

Keywords: orthographic parallel projection, spreadsheets, 3D graphics

3D Graphics with Spreadsheets

Abstract

In the article, the formulas for orthographic parallel projection of 3D bodies on computer screen are derived using secondary school vector algebra. The spreadsheet implementation is demonstrated in six applications that project bodies with increasing intricacy – a convex body (cube) with non-solved visibility, convex bodies (cube, chapel) with solved visibility, a coloured convex body (chapel) with solved visibility, and a coloured non-convex body (church) with solved visibility. The projections are revolvable in horizontal and vertical plane, and they are changeable in size. The examples show an unusual way of using spreadsheets as a 3D computer graphics tool. The applications can serve as a simple introduction to the general principles of computer graphics, to the graphics with spreadsheets, and as a tool for exercising stereoscopic vision. The presented approach is usable at visualising 3D scenes within some topics of secondary school curricula as solid geometry (angles and distances of lines and planes within simple bodies) or analytic geometry in space (angles and distances of lines and planes in E_3), and even at university level within calculus at visualising graphs of $z = f(x,y)$ functions. Examples are pictured.

Keywords: orthographic parallel projection, spreadsheets, 3D graphics

1. Introduction

During its history, spreadsheet has become a powerful tool in economics, technique, science, and school [1]. From a simple environment for performing boring statistical calculations, it has developed to a tool for dynamic graphing, modelling, and simulation, which makes it a profitably environment for teaching and learning mathematics [2], [3] and sciences [4], [5]. In journal Spreadsheets in Education, there are many wonderful articles published about the use of a spreadsheet as a graphing tool (e.g. [6], [7]).

It is natural that having mastered 2D modelling with spreadsheets there are attempts to use it as a 3D modelling tool, as well. It is easy to find on Internet various demonstrations of using spreadsheets as a 3D graphics tool. Most of the applications are simple PC games or just amusements with no or symbolic 3D effect [8 – 12]. In [13], a rendering engine for Microsoft Excel is presented that renders using the spreadsheet cells as a pixel array. By making the cells tiny and turning off the grid, the spreadsheet can be used for basic 3D rendering. The same approach is demonstrated in [14] as an Excel rendering subsystem that the author calls Excel-native Cell Graphics (ECG). Besides that, the author demonstrates another rendering subsystem of Excel that is based on drawing Excel graphic objects on a sheet as on a canvas, which the author calls Office-level Graphics Abstraction Layer (OGAL). The article has aroused a large discussion, and it is referenced in many Internet links. The body is a coloured cube projected in central projection on a sheet, and it is created from triangles using AddPolyline method in VBA.

In this article, other possibilities of creating projections of a body in Excel are shown, even if the body is more complicated and non-convex. Orthographic parallel projection is used, the principle of which is simple enough to derive the transformation equations even using secondary school vector algebra. First, the equations are derived, and then the spreadsheet implementation is demonstrated in six applications. They project bodies with increasing intricacy – a convex body (cube) with non-solved visibility, convex bodies (cube, chapel) with solved visibility, a coloured convex body (chapel), and a coloured non-convex body (church). The projections are revolvable in horizontal and vertical plane. All of them are changeable in size. The uncoloured bodies are made from xy graphs. The coloured bodies are made from freeform shapes. The coordinates of the shape nodes are calculated in cells. Two ways of shape updating are shown. In the first one, the shapes are put on the chart area from Draw panel beforehand, and the node positions are adjusted. In the second one, the shapes are deleted and new ones are created by AddPolyline method (line and fill effects in Excel charts using VBA see in [15], using Excel freeform shapes and conditional colouring in geography see in [16]). The applications can serve as a simple introduction to the general principles of computer graphics, to the graphics with spreadsheets, and as a tool for exercising stereoscopic vision.

2. Orthographic parallel projection

Suppose a point A in space and a plane ρ that is not running through the point. If we make a line through the point perpendicularly to the plane, then the intersection of the line and plane is the orthographic parallel projection of the point onto the plane. If we identify the plane with the computer screen, then we can project 3D objects on it this way. The key problem is to derive the transformation equations between 3D coordinates (x,y,x) of the point in $(\mathrm{O,x,y,z})$ coordinate system, and 2D coordinates (x'',y'') of the projection in screen $(\mathrm{O'',x'',y''})$ coordinate system (the coordinates are in italics while the coordinate axes are in normal font to distinguish). First, we find coordinates (x',y',z') of the projection in $(\mathrm{O,x,y,z})$ system, i.e. the relation $(x,y,z)\rightarrow(x',y',z')$, then we find coordinates (x'',y'') of the projection in $(\mathrm{O'',x'',y''})$ system, i.e. the relation $(x',y',z')\rightarrow(x'',y'')$. We note that (x,y,x) and (x',y',z') are the coordinates of two various points – point A and its projection, while (x',y',z') and (x'',y'') are two various coordinate of the same point – the projection. To make the derivation of the final relation $(x,y,z)\rightarrow(x'',y'')$ easier, we put plane ρ in unit distance from origin O of coordinate system $(\mathrm{O,x,y,z})$, and we take the projection of point O for the origin $\mathrm{O''}$ of coordinate system $(\mathrm{O'',x'',y''})$. If we want to project a body on the screen from various sides (directions), we have to either rotate the body in from of the static plane, or move the plane around the static body. We choose the latter possibility. First, we put the plane parallel with yz plane of $(\mathrm{O,x,y,z})$ system, i.e. perpendicular to x axis. As the distance from origin O is 1 (see above), the equation of the plane is $x=1$; point $\mathrm{O''}$ merge with point $(1,0,0)$. This is the "basic position". Let $\mathrm{x''}$ be the intersection line between plane ρ and coordinate plane xy, $\mathrm{y''}$ be the intersection line between plane ρ and coordinate plane xz. Then, line $\mathrm{x''}$ is parallel to axis y, line $\mathrm{y''}$ is parallel to axis z, and $\mathrm{O''}$ is the intersection of lines x'' and y''. We copy the scale of y and z axes onto lines x'', y''. This way, we get a two dimensional orthogonal coordinate system (O'',x'',y'') in plane ρ.

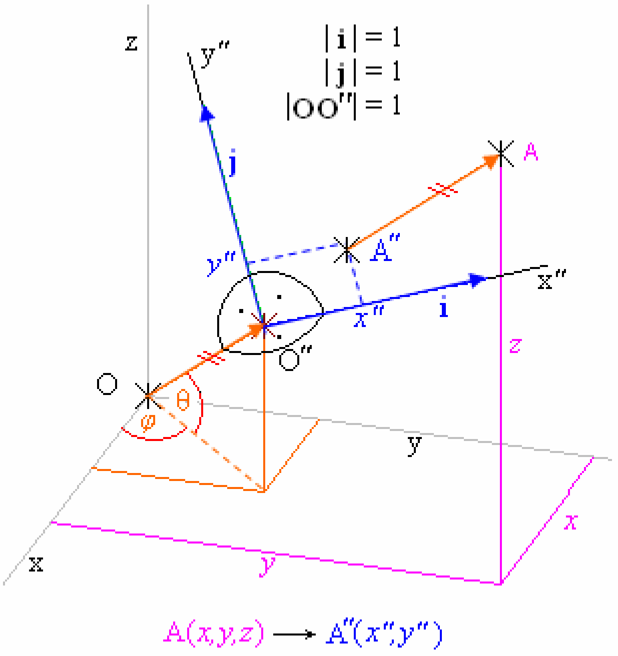


Figure 1: Orthographic projection A'' of point A from system (O,x,y,z) to (O'',x'',y'')

Let point $A''(x'',y'')$ be the orthographic parallel projection of point $A(x,y,z)$. In the basic position (see above), A'' is the front projection of point A, and it holds that $x''=y$, $y''=z$. If we turn plane ρ around axis z so that point O'' merge with point $(0,-1,0)$, then point A'' is the left projection of point A, and it holds that $x''=x$, $y''=z$ (point O'' moved in plane xy to the left at 90° along a quarter-circle). Now, if we turn plane ρ from this position around axis x so that point O'' merge with point $(0,0,1)$, then it holds that $x''=x$, $y''=y$ (point O'' moved in plane yz up at 90° along a quarter-circle).

To get plane ρ to a general position, we turn plane ρ from the basic position around axis z at angle ϕ so that point O'' moved in plane xy along a circular arc. Angle ϕ between semi-axis x^+ and vector $\mathbf{OO''}$ is in plane xy, and we take it positive if point O'' is moving towards semi-axis y^+, otherwise, we take it negative. Afterwards, we turn plane ρ up or down at angle θ so that point O'' moved in plane zO'' along a circular arc. Angle θ between semi-axis z^+ and vector $\mathbf{OO''}$ is in plane zO'', and we take it positive if point O'' is moving towards semi-axis z^+, otherwise, we take it negative (we note that plane zO'' is perpendicular to coordinate plane xy). In fact, angles ϕ and θ are the spherical coordinates of point $\mathbf{O''}$ (similar to terrestrial longitude and latitude). Then, the orthographic parallel projection $A''(x'',y'')$ of point $A(x,y,z)$ onto plane ρ is governed by the equations

$$x''=-x\sin\phi+y\cos\phi, \tag{1}$$

$$y''=-x\sin\theta\cos\phi-y\sin\theta\sin\phi+z\cos\theta. \tag{2}$$

The derivation:
The coordinates of point O'' in system $(\mathrm{O}, \mathrm{x}, \mathrm{y}, \mathrm{z})$ are $(\cos\theta\cos\phi, \cos\theta\sin\phi, \sin\theta)$, thus it holds for vector $\mathbf{OO''}$ that

$$\mathbf{OO''} = (\cos\theta\cos\phi, \cos\theta\sin\phi, \sin\theta). \tag{3}$$

Vector $\mathbf{OO''}$ is the unit normal vector $\mathbf{n}_\varrho$ to plane ρ, i.e. $\mathbf{n}_\varrho = \mathbf{OO''}$. The equation of plane ρ is then

$$\rho : x\cos\theta\cos\phi + y\cos\theta\sin\phi + z\sin\theta - 1 = 0 \tag{4}$$

as it is in unit distance from origin O.

Let **i** be the unit vector of axis x'', and **j** be the unit vector of axis y'' (see Figs. 1, 2). Let α be the angle between vectors **i** and $\mathbf{O''A''}$, β be the angle between vectors **j** and $\mathbf{O''A''}$. Then,

$$x'' = |\mathbf{O''A''}|\cos\alpha = \mathbf{i}\cdot\mathbf{O''A''} = \mathbf{i}\cdot\mathbf{OA''} - \mathbf{i}\cdot\mathbf{OO''} = \mathbf{i}\cdot\mathbf{OA''}, \tag{5}$$

because $\mathbf{i} \perp \mathbf{OO''}$, i.e. $\mathbf{i}\cdot\mathbf{OO''} = 0$. As $\mathbf{O''A''} = \mathbf{OA} + k\cdot\mathbf{OO''}$, where k is a real number, then

$$x'' = \mathbf{i}\cdot\mathbf{OA} + \mathbf{i}\cdot(k\mathbf{OO''}) = \mathbf{i}\cdot\mathbf{OA} + k(\mathbf{i}\cdot\mathbf{OO''}) = \mathbf{i}\cdot\mathbf{OA}\,. \tag{6}$$

Similarly,

$$y'' = |\mathbf{O''A''}|\cos\beta = \mathbf{j}\cdot\mathbf{O''A''} = \mathbf{j}\cdot\mathbf{OA''} - \mathbf{j}\cdot\mathbf{OO''} = \mathbf{j}\cdot\mathbf{OA''} = \mathbf{j}\cdot\mathbf{OA} + \mathbf{j}\cdot(k\mathbf{OO''}) = \mathbf{j}\cdot\mathbf{OA}, \tag{7}$$

because $\mathbf{j} \perp \mathbf{OO''}$ i.e. $\mathbf{j}\cdot\mathbf{OO''} = 0$.

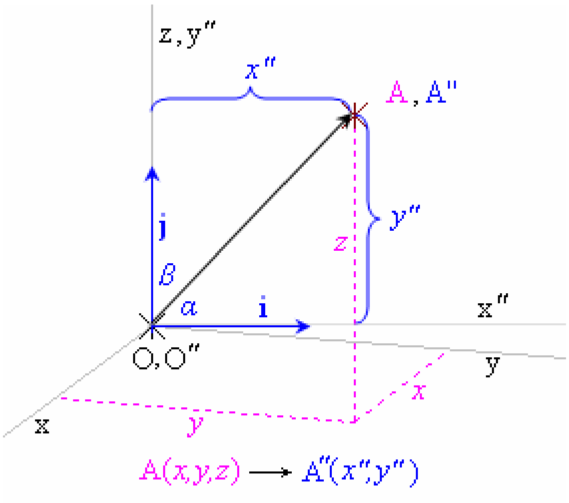


Figure 2: Coordinate system (O''x''y'') on computer screen

Let the coordinates of vector **i** in the 3D coordinate system be $\mathbf{i}(a, b, c)$. It holds that:

a) Vector **i** is parallel to coordinate plane xy at any ϕ, θ, which means that its z coordinate is zero, i.e. $\mathbf{i} = (a, b, 0)$;

b) The size of vector **i** is 1, i.e. $a^2 + b^2 = 1$.

c) Vector **i** is perpendicular to vector $\mathbf{OO''}$, i.e. $\mathbf{i} \cdot \mathbf{OO''} = a\cos\theta\cos\phi + b\cos\theta\sin\phi = 0$, from which $a = -b\tan\phi$.

The system $a^2 + b^2 = 1$, $a = -b\tan\phi$ gives $b = \pm\cos\phi$. If $\phi = 0°$, then $\mathbf{i} = (0,1,0)$, i.e. $b = 1$. Thus, $b = \cos\phi$. Afterwards, $a = -\sin\phi$, and

$$\mathbf{i} = (-\sin\phi,\ \cos\phi,\ 0). \tag{8}$$

It holds for unit vector **j** that $\mathbf{j} = \mathbf{OO''} \times \mathbf{i}$, thus

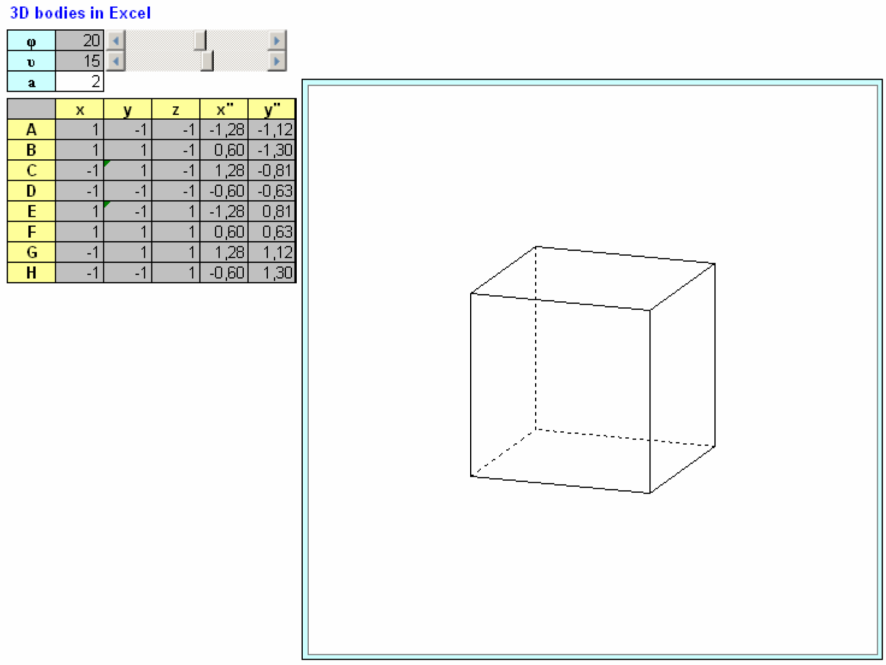
$$\mathbf{j} = (-\sin\theta\cos\phi,\ -\sin\theta\sin\phi,\ \cos\theta). \tag{9}$$

Substituting Eq. (8) to Eq. (6) gives Eq. (1), and substituting Eq. (9) to Eq. (7) gives Eq. (2). Note that if $\phi = -90°$ and $\theta = 90°$, then Eqs. (1), (2) give $x''_{\mathrm{A}} = x_{\mathrm{A}}$, $y''_{\mathrm{A}} = y_{\mathrm{A}}$.

More about Orthographic projection see e.g. in [17].

3. Projection of a simple convex body with non-solved visibility

Let A – H be the vertices of a cube of edge a = 2 that is situated in a coordinate system with faces parallel to the coordinate planes, and with the intersection of body diagonals at the origin of the coordinate system. Then, the coordinates x, y, z of the vertices are easily calculable, and transformable afterwards to x'', y'' using Eqs. (1, 2).



3D bodies in Excel

φ	20
υ	15
a	2

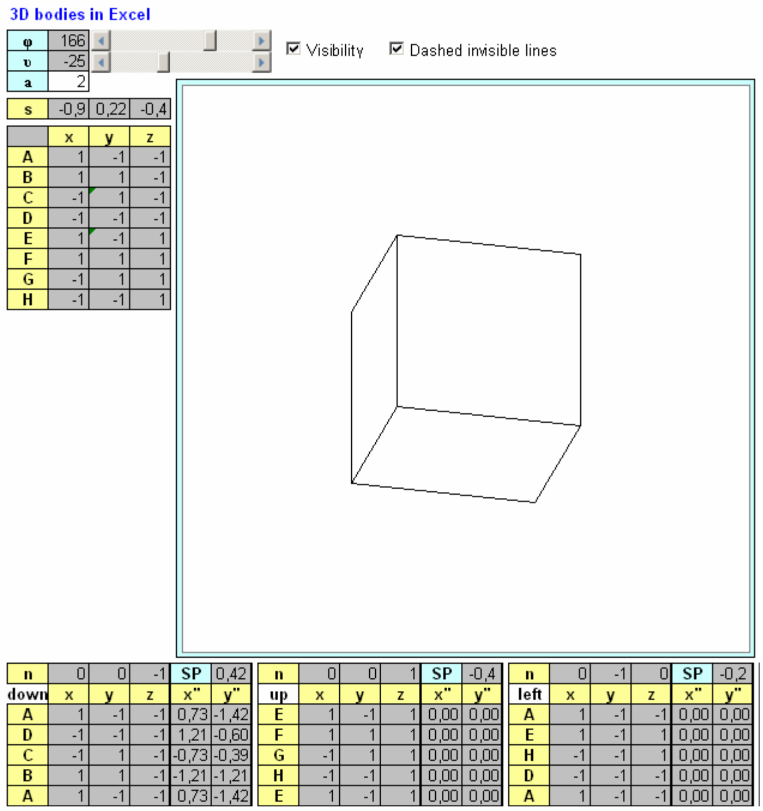
	x	y	z	x"	y"
A	1	-1	-1	-1,28	-1,12
B	1	1	-1	0,60	-1,30
C	-1	1	-1	1,28	-0,81
D	-1	-1	-1	-0,60	-0,63
E	1	-1	1	-1,28	0,81
F	1	1	1	0,60	0,63
G	-1	1	1	1,28	1,12
H	-1	-1	1	-0,60	1,30

Figure 3: Cube with non-solved visibility

The spreadsheet application is in Fig. 3. Angles ϕ, θ are only changeable by the scrollbars by 1°. Length a is changeable by hand only, which changes the size of the cube. The cube is drawn using 12 two-point xy graphs (one graph for each edge). The names of the series as well as the names of the vertices are cleared away. The invisible edges are made dotted by hand. If they become visible when rotating the cube, the user has to change them to full by double-click. No macro is used in the application.

4. Projection of a simple convex body with solved visibility

If the body is convex, then the visibility of a face is solvable using its normal vector **n** that is oriented out of the body. If the angle between vector **n** and vector $\mathbf{OO''}$ is bigger than 90°, then the face is invisible. That means that if $\mathbf{n} \cdot \mathbf{OO''} \geq 0$, then the face is visible. Let the body be the cube from section 3. The coordinates of the face normal vectors are clear from the position of the cube in the 3D coordinate system, e.g. for the front face it is (1,0,0).



3D bodies in Excel

φ	166
υ	-25
a	2

☑ Visibility ☑ Dashed invisible lines

s	-0,9	0,22	-0,4

	x	y	z
A	1	-1	-1
B	1	1	-1
C	-1	1	-1
D	-1	-1	-1
E	1	-1	1
F	1	1	1
G	-1	1	1
H	-1	-1	1

n	0	0	-1	SP	0,42
down	x	y	z	x"	y"
A	1	-1	-1	0,73	-1,42
D	-1	-1	-1	1,21	-0,60
C	-1	1	-1	-0,73	-0,39
B	1	1	-1	-1,21	-1,21
A	1	-1	-1	0,73	-1,42

n	0	0	1	SP	-0,4
up	x	y	z	x"	y"
E	1	-1	1	0,00	0,00
F	1	1	1	0,00	0,00
G	-1	1	1	0,00	0,00
H	-1	-1	1	0,00	0,00
E	1	-1	1	0,00	0,00

n	0	-1	0	SP	-0,2
left	x	y	z	x"	y"
A	1	-1	-1	0,00	0,00
E	1	-1	1	0,00	0,00
H	-1	-1	1	0,00	0,00
D	-1	-1	-1	0,00	0,00
A	1	-1	-1	0,00	0,00

Figure 4: Cube with switched on visibility

The spreadsheet application is in Fig. 4. In the rows below the chart, there are 6 ranges that define each face (3 of them pictured). The vertices of the faces are written by hand anticlockwise if looking at the face from outside of the cube (this is not necessary in this case; the reason see in Section 5). The x, y, z coordinates of each vertex are returned by VLOOKUP function from the list of vertices that is on the left from the chart. Above that, there are the coordinates of vector $\mathbf{s} = \mathbf{OO''}$. The coordinates of the face normal vectors are written by hand. The scalar product $\mathbf{n} \cdot \mathbf{s}$ is calculated by SUMPRODUCT function (abbreviation SP). Coordinates x'', y'' of the vertices are calculated using Eqs. (1, 2), however, IF the scalar product (SP) is less than zero, then x'', y'' are set to zero. This causes the invisible face to collapse in point (0,0). The IF function is nested in another IF function that interprets the Boolean content of cell M12 (hidden under the chart) linked with CheckBox1 (captioned "Visible").

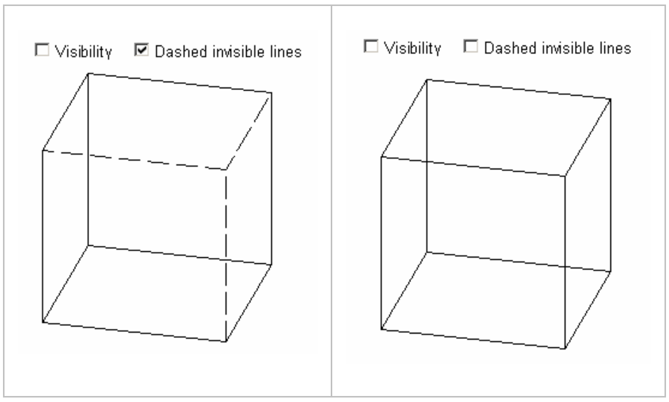


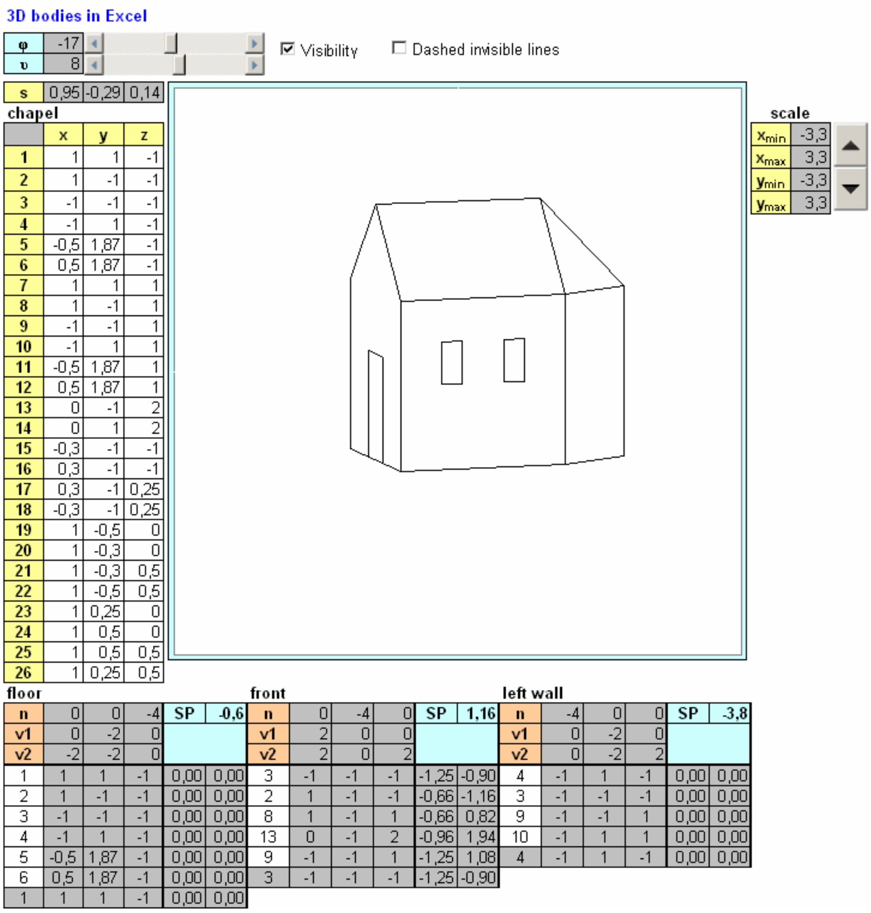
Figure 5: Cube with switched off visibility

Each face is drawn separately as a five-point xy graph where the fifth point merges with the first one. The dash dot or continuous (i.e. full) character of the borders of the invisible faces is governed by a macro that is a method of the scrollbar for angle ϕ. If the content of cell M13 (hidden under the chart) linked with CheckBox2 (captioned "Dashed invisible lines") is TRUE and the corresponding SP is negative, then the face border is dash dotted, otherwise, it is full. This evaluation is made for each face. The application is in Figs. 4, 5.

5. Projection of a convex body with solved visibility

The situations is rather complicated if the body is irregular, e.g. if it is a chapel (Fig. 6). Then, the outgoing normal vectors have to be calculated as the vector product of two non-collinear vectors in the face. In the rows below the chart, there are 15 ranges that define each face (3 of them pictured). The vertices of the faces are written by hand anticlockwise if looking at the face from outside of the cube. Vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ in a face are defined as the difference between the second and first, and the third and first vertex of the vertex list of the face. Then, vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ are non-collinear, and the vector product of them gives the outgoing normal vector of the face. The dash dot or full character of the borders of the invisible faces is solved in the same way as in section 4.

The scale cells on the right from the chart refer to cell M14 that is linked with the spinbutton and that is hidden under the chart. The axis ranges are changed by a macro that is a method of the spinbutton. Clicking on the spinbutton changes the range of the xy coordinate system which causes enlarging or diminishing the chapel. The application is in Figs. 6, 7.



3D bodies in Excel

φ	-17
υ	8

☑ Visibility □ Dashed invisible lines

s	0,95	-0,29	0,14

chapel

	x	y	z
1	1	1	-1
2	1	-1	-1
3	-1	-1	-1
4	-1	1	-1
5	-0,5	1,87	-1
6	0,5	1,87	-1
7	1	1	1
8	1	-1	1
9	-1	-1	1
10	-1	1	1
11	-0,5	1,87	1
12	0,5	1,87	1
13	0	-1	2
14	0	1	2
15	-0,3	-1	-1
16	0,3	-1	-1
17	0,3	-1	0,25
18	-0,3	-1	0,25
19	1	-0,5	0
20	1	-0,3	0
21	1	-0,3	0,5
22	1	-0,5	0,5
23	1	0,25	0
24	1	0,5	0
25	1	0,5	0,5
26	1	0,25	0,5

scale

x_{min}	-3,3
x_{max}	3,3
y_{min}	-3,3
y_{max}	3,3

floor

n	0	0	-4	SP	-0,6
v1	0	-2	0		
v2	-2	-2	0		
1	1	1	-1	0,00	0,00
2	1	-1	-1	0,00	0,00
3	-1	-1	-1	0,00	0,00
4	-1	1	-1	0,00	0,00
5	-0,5	1,87	-1	0,00	0,00
6	0,5	1,87	-1	0,00	0,00
1	1	1	-1	0,00	0,00

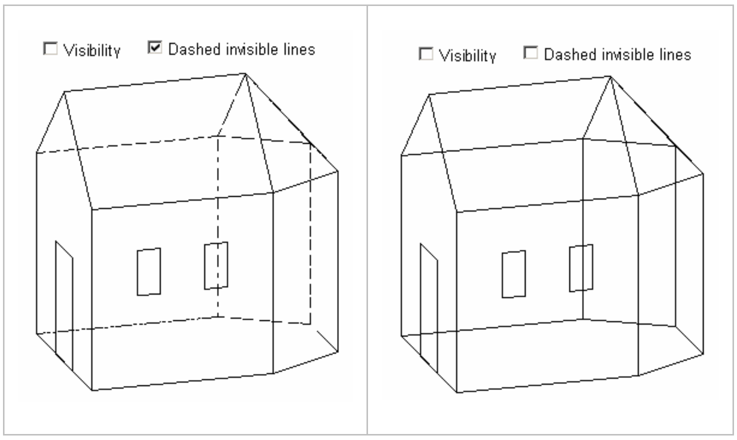
front

n	0	-4	0	SP	1,16
v1	2	0	0		
v2	2	0	2		
3	-1	-1	-1	-1,25	-0,90
2	1	-1	-1	-0,66	-1,16
8	1	-1	1	-0,66	0,82
13	0	-1	2	-0,96	1,94
9	-1	-1	1	-1,25	1,08
3	-1	-1	-1	-1,25	-0,90

left wall

n	-4	0	0	SP	-3,8
v1	0	-2	0		
v2	0	-2	2		
4	-1	1	-1	0,00	0,00
3	-1	-1	-1	0,00	0,00
9	-1	-1	1	0,00	0,00
10	-1	1	1	0,00	0,00
4	-1	1	-1	0,00	0,00

Figure 6: Chapel with switched on visibility; axis range from -3.3 to 3.3



□ Visibility ☑ Dashed invisible lines

□ Visibility □ Dashed invisible lines

Figure 7: Chapel with switched off visibility; axis range from -3 to 3

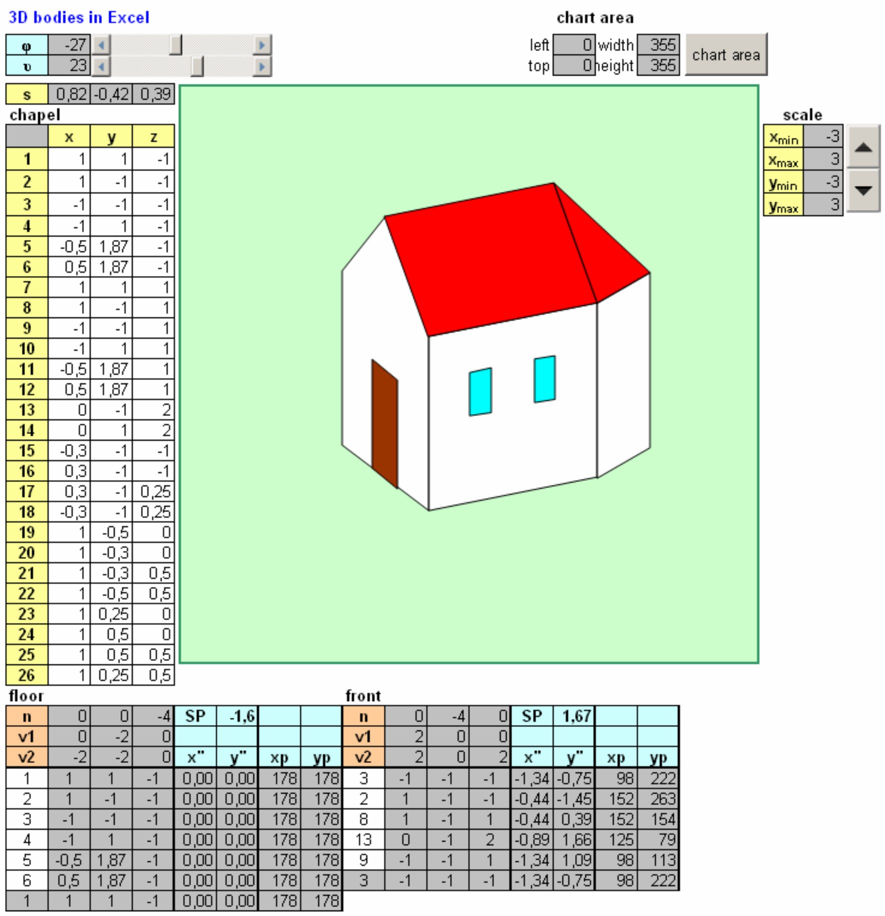
6. Projection of a coloured convex body with solved visibility

To get the chapel coloured is impossible using graphs. Completely different approach is necessary. That is based on setting up the projection of the body from shapes placed in a sheet or a chart. Both charts and sheets have they inner planar pixel coordinate system that origins at the left upper corner. Here, we use chart area for drawing the projection. Remark: Distinguish between the chart area and the plot area. The plot area is embedded in the chart area, and it can be moved and resized within that as well as deleted from that, which has been done in the charts in our applications. Here, the chart area only serves as a background for building up the body from shapes. The *width* and *height* of the chart area is returned in pixels by a macro that is a method of button "chart area". The pixels coordinates of a point in the chart area are labelled $(x_{\mathrm{p}}, y_{\mathrm{p}})$. The transformation from (x'', y'') to $(x_{\mathrm{p}}, y_{\mathrm{p}})$ is governed by the formulas

$$x_{\mathrm{p}} = width\,(x - x_{\min})/(x_{\max} - x_{\min}), \tag{10}$$

$$y_{\mathrm{p}} = height\,(y_{\max} - y)/(y_{\max} - y_{\min}). \tag{11}$$

The shapes that constitute the body are freeforms that we had put on the chart area from Draw panel as "protoforms" (see them in Fig. 9 at the bottom, right, and top sides). They have the required number of nodes and the required interior and border colours. The coordinates $(x_{\mathrm{p}}, y_{\mathrm{p}})$ of the nodes are calculated in columns xp, yp . The positions of the nodes are adjusted by a VBA macro that is a method of the scrollbar for angle ϕ. The first click on a scrollbar or spinbutton changes the protoforms to the correct ones that form up the projection of the body. The forms stay in the chart area all the time, only their position is changes via adjusting the node coordinates.



3D bodies in Excel

φ	-27
υ	23

chart area

left 0 width 355

top 0 height 355

s	0,82	-0,42	0,39

chapel

	x	y	z
1	1	1	-1
2	1	-1	-1
3	-1	-1	-1
4	-1	1	-1
5	-0,5	1,87	-1
6	0,5	1,87	-1
7	1	1	1
8	1	-1	1
9	-1	-1	1
10	-1	1	1
11	-0,5	1,87	1
12	0,5	1,87	1
13	0	-1	2
14	0	1	2
15	-0,3	-1	-1
16	0,3	-1	-1
17	0,3	-1	0,25
18	-0,3	-1	0,25
19	1	-0,5	0
20	1	-0,3	0
21	1	-0,3	0,5
22	1	-0,5	0,5
23	1	0,25	0
24	1	0,5	0
25	1	0,5	0,5
26	1	0,25	0,5

scale

x_{min}	-3
x_{max}	3
y_{min}	-3
y_{max}	3

floor

n	0	0	-4	SP	-1,6		
v1	0	-2	0				
v2	-2	-2	0	x"	y"	xp	yp
1	1	1	-1	0,00	0,00	178	178
2	1	-1	-1	0,00	0,00	178	178
3	-1	-1	-1	0,00	0,00	178	178
4	-1	1	-1	0,00	0,00	178	178
5	-0,5	1,87	-1	0,00	0,00	178	178
6	0,5	1,87	-1	0,00	0,00	178	178
1	1	1	-1	0,00	0,00	178	178

front

n	0	-4	0	SP	1,67		
v1	2	0	0				
v2	2	0	2	x"	y"	xp	yp
3	-1	-1	-1	-1,34	-0,75	98	222
2	1	-1	-1	-0,44	-1,45	152	263
8	1	-1	1	-0,44	0,39	152	154
13	0	-1	2	-0,89	1,66	125	79
9	-1	-1	1	-1,34	1,09	98	113
3	-1	-1	-1	-1,34	-0,75	98	222

Figure 8: Chapel with colours

7. Projection of a coloured non-convex body with solved visibility

The protoforms for the church are in Fig. 9. The first click on a scrollbar changes them to the correct shapes.

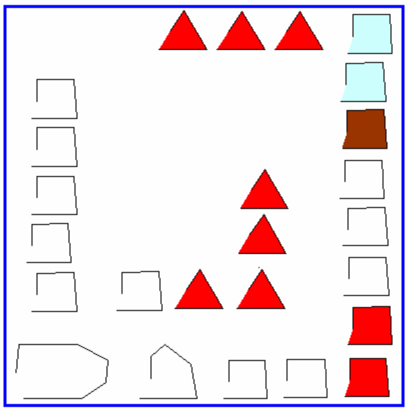
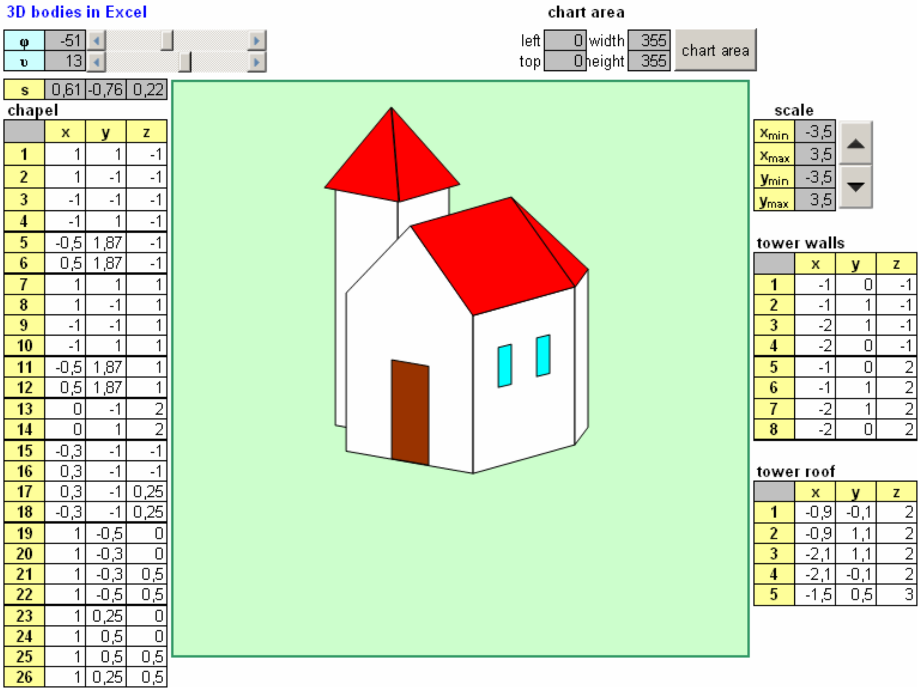


Figure 9: Church protoforms

However, as the body is non-convex, the problem of overlapping of the parts arises. That is solved by dividing the body into convex parts, which are the chapel, the tower without the roof, and the tower roof. Each of them is drawn by its own subroutine. From the main program, which is called when the scrollbars or spinbutton change, the subroutines are called successively according to the actual overlapping and visibility of the convex components of the church. The overlapping follows from the projection direction which is a function of angles ϕ and θ. The application is in Figs. 10, 11. The sheet is protected without a password. All the cells and objects are locked. Still, the body rotation works. That is because the position of the shape nodes is changed only.



3D bodies in Excel

φ	-51
υ	13

chart area

left	0	width	355
top	0	height	355

s	0,61	-0,76	0,22

chapel

	x	y	z
1	1	1	-1
2	1	-1	-1
3	-1	-1	-1
4	-1	1	-1
5	-0,5	1,87	-1
6	0,5	1,87	-1
7	1	1	1
8	1	-1	1
9	-1	-1	1
10	-1	1	1
11	-0,5	1,87	1
12	0,5	1,87	1
13	0	-1	2
14	0	1	2
15	-0,3	-1	-1
16	0,3	-1	-1
17	0,3	-1	0,25
18	-0,3	-1	0,25
19	1	-0,5	0
20	1	-0,3	0
21	1	-0,3	0,5
22	1	-0,5	0,5
23	1	0,25	0
24	1	0,5	0
25	1	0,5	0,5
26	1	0,25	0,5

scale

x_{min}	-3,5
x_{max}	3,5
y_{min}	-3,5
y_{max}	3,5

tower walls

	x	y	z
1	-1	0	-1
2	-1	1	-1
3	-2	1	-1
4	-2	0	-1
5	-1	0	2
6	-1	1	2
7	-2	1	2
8	-2	0	2

tower roof

	x	y	z
1	-0,9	-0,1	2
2	-0,9	1,1	2
3	-2,1	1,1	2
4	-2,1	-0,1	2
5	-1,5	0,5	3

Figure 10: Church

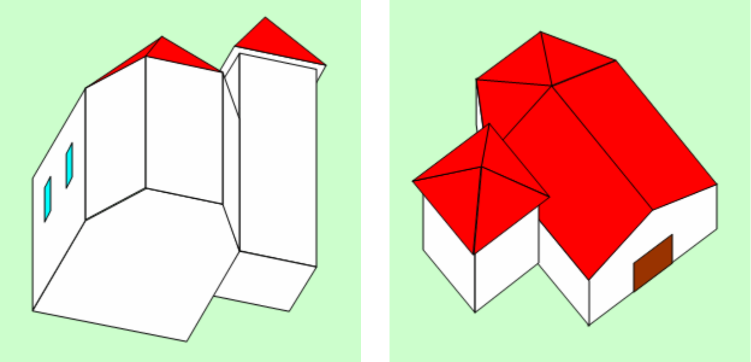
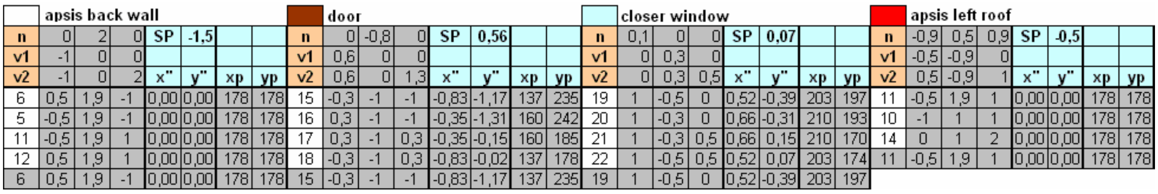


Figure 11: Other views of the church



□	apsis back wall							
n	0	2	0	SP	-1,5			
v1	-1	0	0					
v2	-1	0	2	x"	y"	xp	yp	
6	0,5	1,9	-1	0,00	0,00	178	178	
5	-0,5	1,9	-1	0,00	0,00	178	178	
11	-0,5	1,9	1	0,00	0,00	178	178	
12	0,5	1,9	1	0,00	0,00	178	178	
6	0,5	1,9	-1	0,00	0,00	178	178	

🟫	door							
n	0	-0,8	0	SP	0,56			
v1	0,6	0	0					
v2	0,6	0	1,3	x"	y"	xp	yp	
15	-0,3	-1	-1	-0,83	-1,17	137	235	
16	0,3	-1	-1	-0,35	-1,31	160	242	
17	0,3	-1	0,3	-0,35	-0,15	160	185	
18	-0,3	-1	0,3	-0,83	-0,02	137	178	
15	-0,3	-1	-1	-0,83	-1,17	137	235	

🟦	closer window							
n	0,1	0	0	SP	0,07			
v1	0	0,3	0					
v2	0	0,3	0,5	x"	y"	xp	yp	
19	1	-0,5	0	0,52	-0,39	203	197	
20	1	-0,3	0	0,66	-0,31	210	193	
21	1	-0,3	0,5	0,66	0,15	210	170	
22	1	-0,5	0,5	0,52	0,07	203	174	
19	1	-0,5	0	0,52	-0,39	203	197	

🟥	apsis left roof							
n	-0,9	0,5	0,9	SP	-0,5			
v1	-0,5	-0,9	0					
v2	0,5	-0,9	1	x"	y"	xp	yp	
11	-0,5	1,9	1	0,00	0,00	178	178	
10	-1	1	1	0,00	0,00	178	178	
14	0	1	2	0,00	0,00	178	178	
11	-0,5	1,9	1	0,00	0,00	178	178	

Figure 12: Defining shape line and fill colours by cell border and interior colours

The manual making protoforms can be replaced with creating shapes by a subroutine, and that is by using either BuildFreeform or AddPolyline methods of Shapes Collection. After creating the shapes, the projection can operate as it is described in section 6, i.e. by adjusting the shape nodes. However, there is another possibility, and that is by deleting the "old" shapes and creating "new" ones at each change of the view or size, i.e. at each click on a scrollbar or spinbutton. This method is used in the application on sheet "Church Polyline" where AddPolyline method creates the new shapes. We remark that before the first click on a scrollbar or spinbutton, i.e. just after the development of the application when no shape is drawn yet in the chart area (this is already not the case of the presented application) the user have to remove the first line of the main program (by REM) as there is no shape to be deleted. Immediately after the first click, put the line back to delete the old shapes when clicking. In this application, the border and interior colours of the shapes (properties Line.ForeColor and Fill.ForeColor) are set up according to the Borders.Color and Interior.Color of the cells that are designed for this purpose (see them in Fig. 11). This approach enables to change the colours of the shapes even if the chart is locked. In this case, locking the chart requires adding other two commands in the main program code, and that is switching off the sheet protection before deleting and creating the shapes, and switching on the sheet protection at the end of the routine.

In Fig. 12, there is the application with changed colours and with hidden rows and columns to get the input cells visible only.

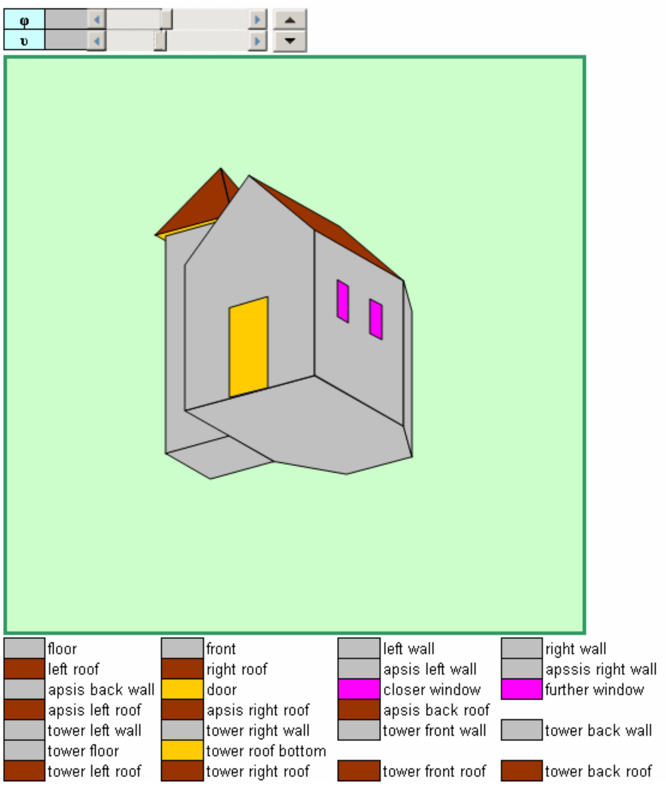


Figure 13: The final interface enabling to change the colours in a locked chart

8. Conclusions

In the article, several ways of using spreadsheets as a 3D computer graphics tool are presented. As it is shown, the problem of creating uncoloured projections of a convex body allowing for the visibility is solvable without programming if using xy graphs. The problem of automatic change of the character of the lines when changing from visible to invisible (or vice versa) as well as the problem of creating coloured projections are only solvable using programming. The coloured projection is based either upon adjusting the nodes of permanent shapes or upon deleting shapes and creating new ones.

The paper and the applications are aimed for teaching informatics (computer graphics + spreadsheets + VBA programming) not mathematics (except for section 2 and the visibility problem). However, the presented approach based upon xy graphs is usable at visualising 3D scenes within some topics of secondary school mathematics as solid geometry (angles and distances of lines and planes within simple bodies, Fig. 14, [3]; the application was successfully tried at grammar school), or analytic geometry in space (angles and distances of lines and planes in E3, Fig. 15, [3]), and even at university level within calculus at drawing graphs of functions of two variables (Fig. 16).

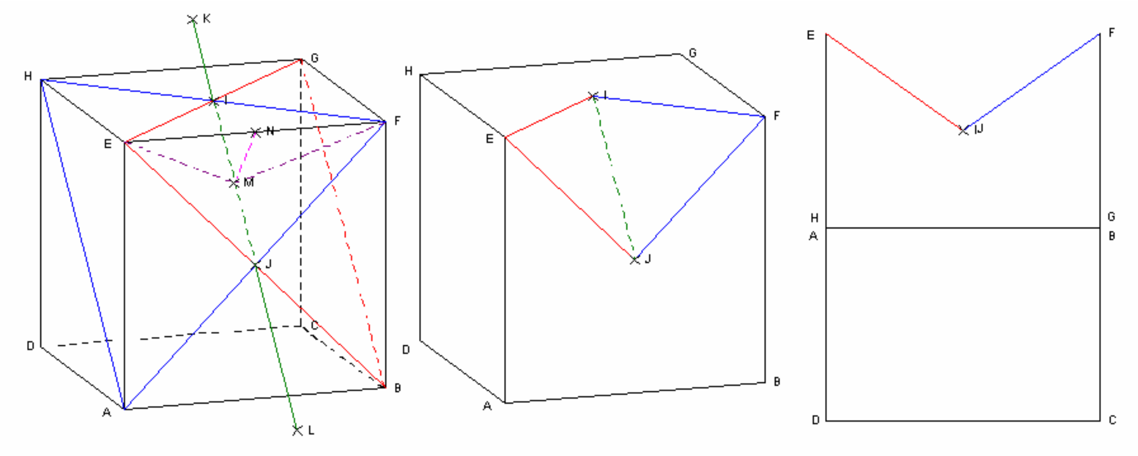


Figure 14: The angle of planes AFH (red) and BEG (blue) in a cube

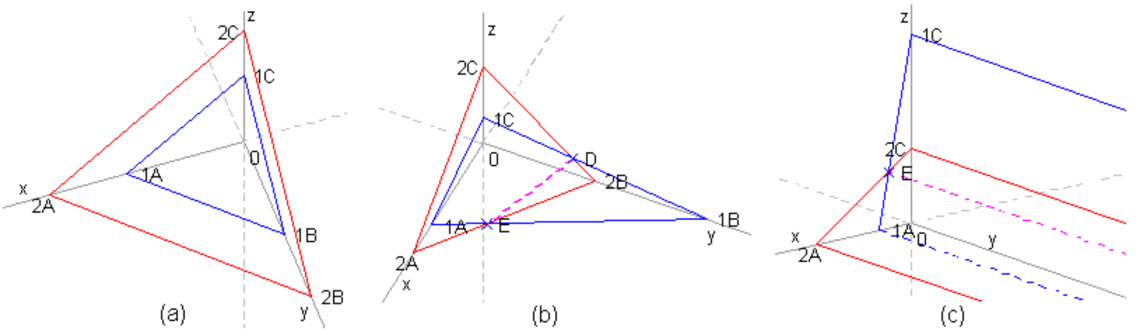


Figure 15: Two planes in E_3 (blue, red): a) $\alpha : x+z+y-3=0$, $\alpha : x+z+y-5=0$; b) $\alpha : 2x+y+6z-6=0$, $\alpha : 3x+4z+4z-12=0$; c) $\alpha : 5x+z-5=0$, $\alpha : 2x+3z-6=0$;

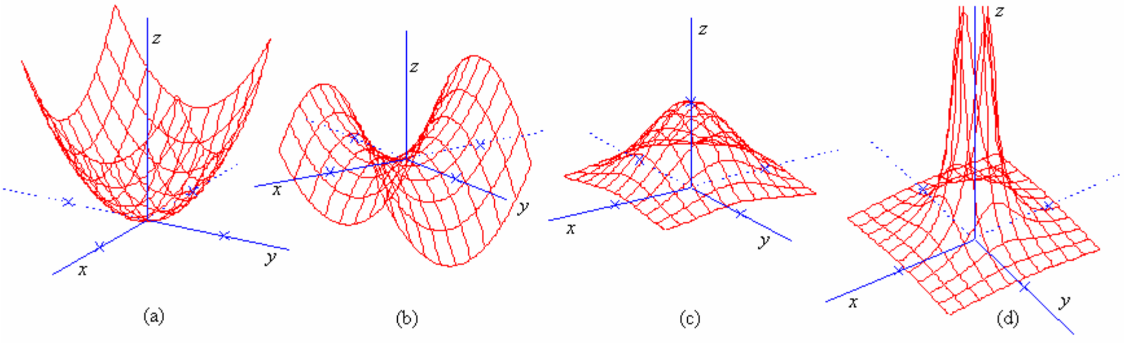


Figure 16: Graphs of: a) $z=x^2+y^2$, b) $z=x^2-y^2$, c) $z=\mathrm{e}^{-2\left(x^2+y^2\right)}$, d) $z=1/\left(x^2+y^2\right)$

References

[1] Baker, John and Sugden, Steve (2003) "Spreadsheets in Education –The First 25 Years," *Spreadsheets in Education (eJSiE)*: Vol. 1: Iss. 1, Article 2
Available at: http://epublications.bond.edu.au/ejsie/vol1/iss1/2

[2] Arganbright, Deane (1993) *Practical Handbook of Spreadsheet Curves and Geometric Constructions*. CRC Press, Boca Raton.

[3] Benacka, Jan and Ceretkova, Sona (2009) "Grammar school mathematics with MS ExcelTM" (poster). In *Proceedings of CERME 6 – Sixth Conference of European Research in Mathematics Education*, Lyon, France, in press. Available online at http://www.ki.fpv.ukf.sk/~jbenacka/download/CERME6/Grammar_school_mathematics_with_Excel.pdf

[4] Neuwirth, Erich and Arganbright, Deane (2004). The Active Modeler: Mathematical Modeling with Microsoft Excel. Brooks/Cole, Belmont.

[5] Benacka, Jan and Stubna, Igor (2009) "Ball launched against an inclined plane – an example of using recurrent sequences in school physics". *International Journal of Mathematical Education in Science and Technology*, iFirst (online). Available at http://www.informaworld.com/smpp/content~content=a910913813~db=all~order=pubdate

[6] Arganbright, Deane (2005) "Enhancing Mathematical Graphical Displays in Excel through Animation," *Spreadsheets in Education (eJSiE)*: Vol. 2: Iss. 1, Article 8. Available at: http://epublications.bond.edu.au/ejsie/vol2/iss1/8

[7] Baker, John (2007) "Excel and the Goldbach Comet," *Spreadsheets in Education (eJSiE)*: Vol. 2: Iss. 2, Article 2.
Available at: http://epublications.bond.edu.au/ejsie/vol2/iss2/2

[8] http://www.excelgames.org

[9] http://www.youtube.com/watch?v=Fmkd9Jv4F1o&feature=related

[10] http://www.youtube.com/watch?v=YAIQZcabtSM&feature=related

[11] http://www.youtube.com/watch?v=ROwTX2YpHAs&feature=related

[12] http://www.youtube.com/watch?v=Zx3yzHPjDD0&feature=related

[13] http://www.etheus.net/XL3DRender

[14] http://www.gamasutra.com/view/feature/3563/microsoft_excel_revolutionary_3d_.php

[15] http://peltiertech.com/Excel/Charts/VBAdraw.html

[16] http://tushar-mehta.com/excel/charts/0301-dashboard-conditional%20shape%20colors.htm

[17] Orthographic projection. In *Wikipedia, the free encyclopedia*, retrieved 12:30, November 24, 2008, from http://en.wikipedia.org/wiki/Orthographic_projection