BSTools

pozwalaj — a session log

Przemysław Kiciak

Version 0.27, June 20, 2011
TEX-processed June 20, 2011

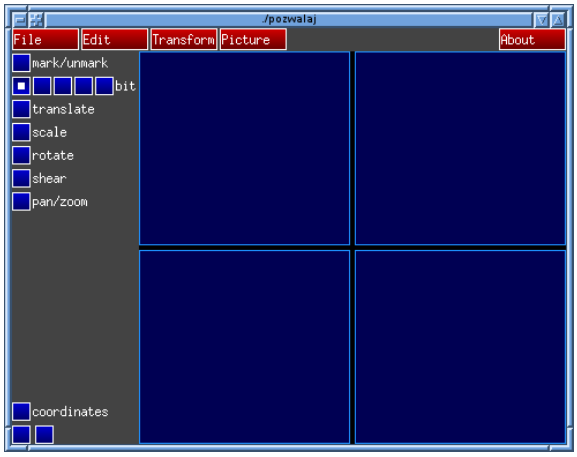
This document contains screen dumps and comments written during a session with the program `pozwalaj`, one of the demonstration programs of the package BSTools. During this session a blending surface has been designed, using the interactive tools of the program and one of the built in shape optimization procedures for such surfaces.

The executable file, `pozwalaj`, by default is located in the directory

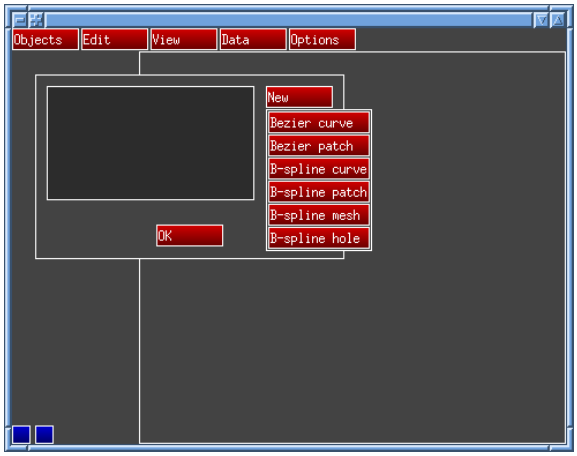
`bstools-0.27/demo/bin/`

where there is also the file `pozwalaj_proc`, containing the shape optimization procedures; as the computations may take a long time, they are performed independently of the interaction provided by the main executable file. The program `pozwalaj_proc` is supposed to be executed only when invoked by the program `pozwalaj` (and run from the command line it will immediately terminate).

After invoking, the program displays two windows (some XWindow managers may place them initially in such a way that one window obscures the other one). The first window displays the geometric data (curves, surfaces and their control polygons and meshes). If 3D objects are displayed, the geometry window is divided into four areas. Three of them show orthogonal projections of the geometric objects onto the xz, yz and xy planes, and one (lower right) shows a perspective projection. A user may change the projection centre by moving the cursor into this area, pressing the left button and moving the cursor.



One of the geometric objects (curves or surfaces, none are present at the beginning of program execution) is distinguished as current. The second window allows the user to make actions specific for the current object, via menus specific for that object. After clicking the button labelled Objects and then New in the popup menu, the window looks like this:

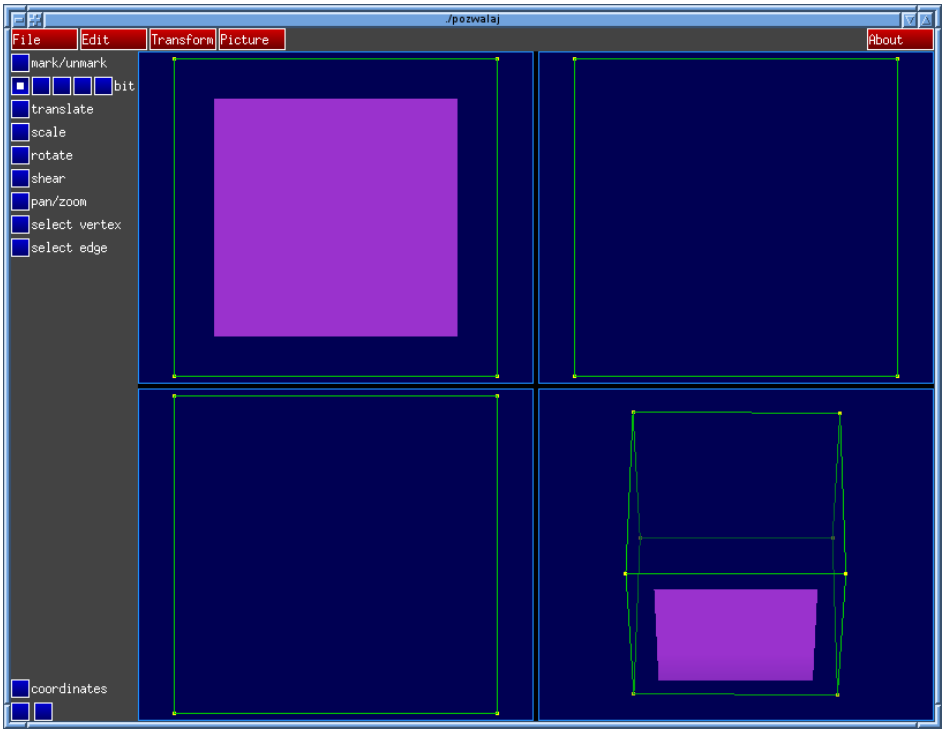


Then, after clicking the button B-spline mesh and then Add in yet another popup, the program creates a new object, which is a spline surface represented by a mesh. Initially this mesh has one facet with four vertices and edges.

After clicking the buttons Data and cube, we choose a mesh, whose facets form the boundary of a cube; the length of its edges is 2. Then clicking the Edit button returns to the menu making it possible to edit the mesh topology. Resizing the window (making it slightly higher) causes all widgets of this menu to fit in.

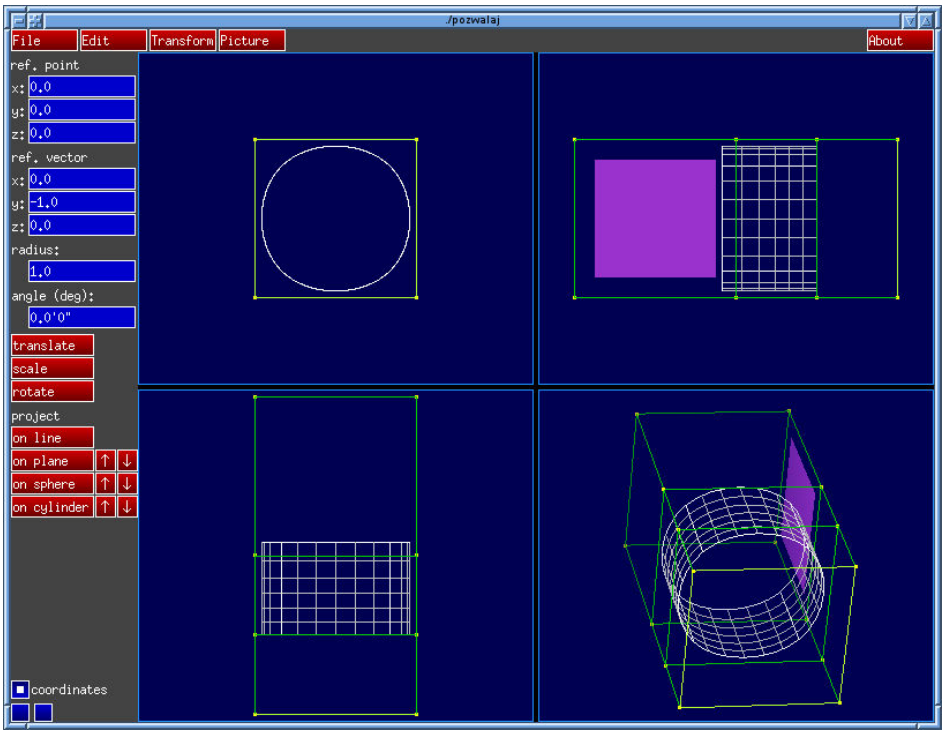
With the cursor back in the first window, on the object images, typing F makes the program find a bounding cube of the object and fit it in the visible area.

Now we edit the mesh. In the second window, click twice (using the left mouse button) the green widget labelled facet (to decrease the number, use the right button, also the mouse wheel works here). This will distinguish the facet number 1, which will be displayed as follows:

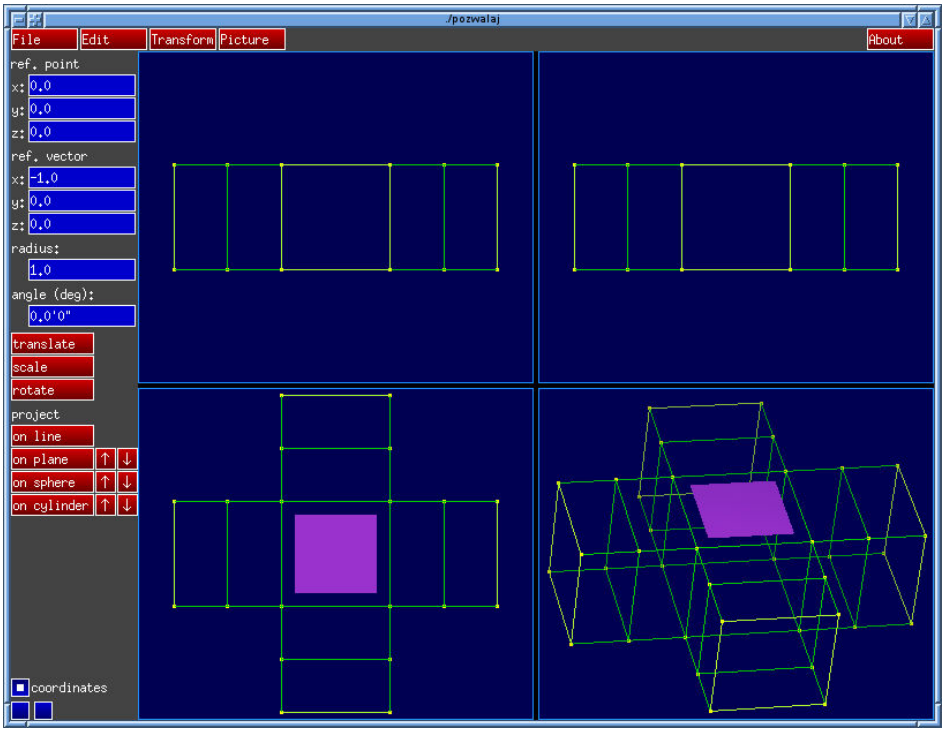


In the second window, click the double edges button. This executes the Eulerian operation, which produces four new facets surrounding the distinguished facet. The new facets are quadrangular, but they are degenerated to line segments.

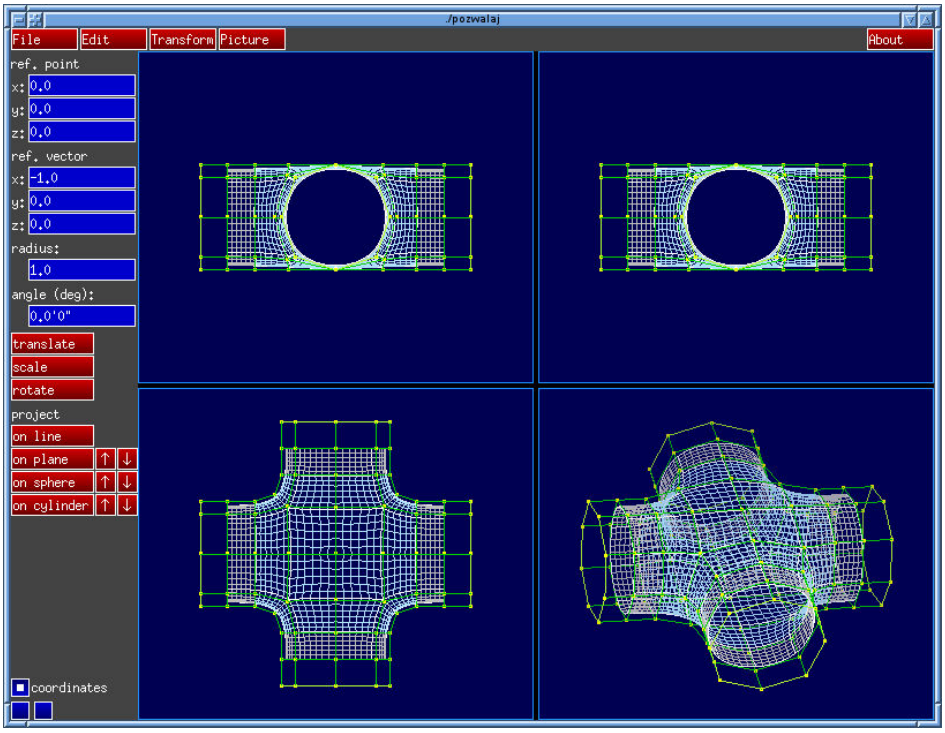
In the first window, click the Transform button. Then, using the text editing widgets (the blue ones, they are activated by clicking on them, and deactivated by clicking aside), type in the coordinates of the reference vector $[0, -1, 0]$, and then click the translate button. Then, in the second window, click double edges and in the first window click translate again. Then click the remove button below the facet number widget. The window now looks as follows:



After removing the facet, another facet became number 1. We click the button double edges, then in the first window we enter the reference vector $[1,0,0]$ and click translate, we double edges and translate once more and again we remove the facet. After removing it, we double the edges of the new facet number one, translate its vertices by the vector $[0,1,0]$, again we double and translate and remove the facet. For the fourth facet, which became number 1, twice we double the edges and translate the vertices by the vector $[-1,0,0]$, then we remove the facet. The result is shown on the next picture:



The mesh has now four closed boundaries, each formed by four edges adjacent to the removed facets. Now, in the second window, we click the refine button. It invokes the procedure implementing the mesh refinement operator, the composition of mesh doubling followed by three averaging operations (actually, the number of averagings is the surface degree, 3 by default). Here is what we obtain:

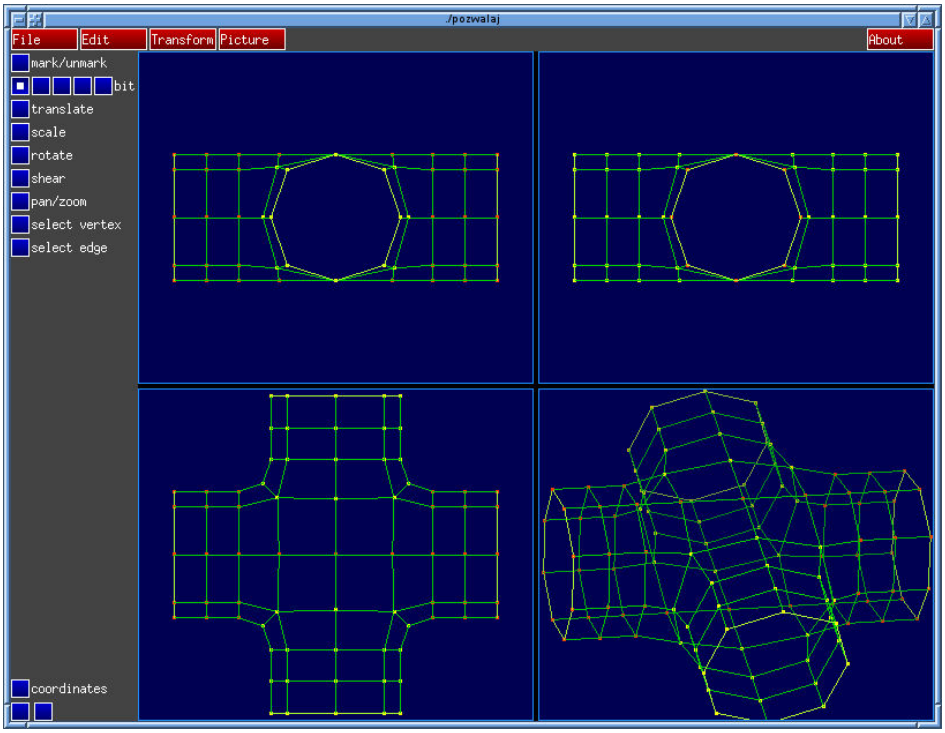


The light grey lines are constant parameter curves of the bicubic patches, corresponding to the regular elements of the surface domain. The binonic patches, represented by special elements of the mesh, are drawn in light blue. For convenience, we may click the View button in the second window and turn off (by clicking) the switches which control displaying the surface (i.e. the bicubic patches) and the hole filling (binonic patches), thus leaving only the mesh vertices and edges on the picture.

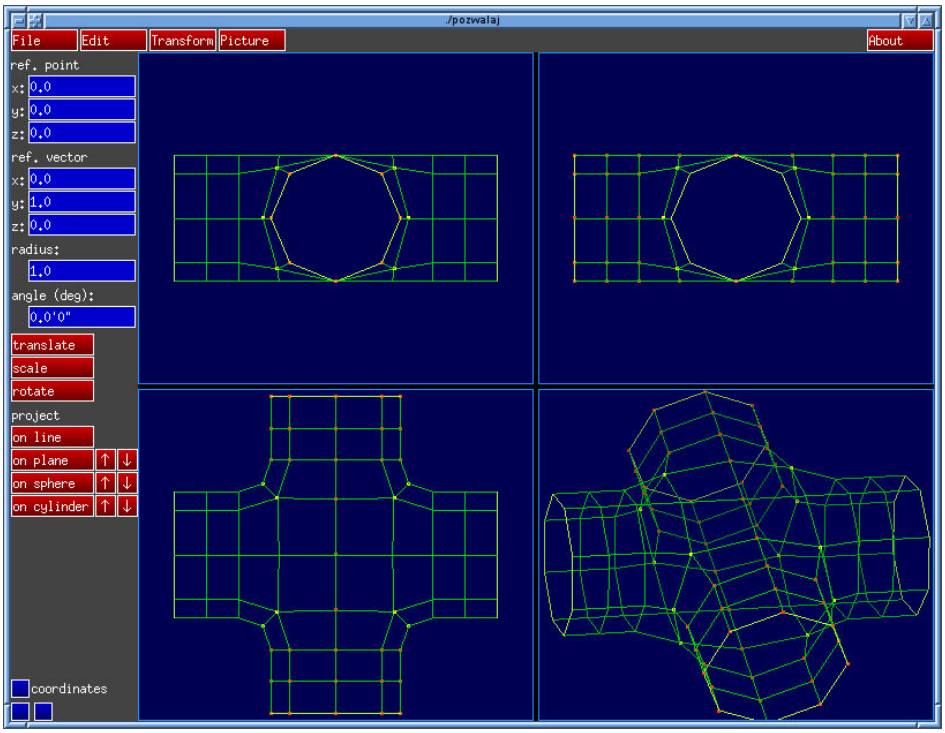
We are going to obtain a blending surface, which is a junction of two crossing cylindric tubes. Of course, bicubic splines cannot represent cylinders of revolution exactly, but if the mesh is dense enough and the vertices are located on a cylinder of revolution, the spline surface may approximate a cylinder with an arbitrarily small error. Therefore in the next step the mesh vertices will be projected onto cylinders. Here is the method: click the Edit button in the top menu of the first window. Then click the mark/unmark switch to turn it on. Now move the cursor to one of the object image windows. Vertices may be marked individually by clicking with

the left mouse button and unmarked by clicking with the right button. Also it is possible to press the button, move the cursor and release the button in order to mark or unmark all vertices in the rectangular area indicated by this mouse movement.

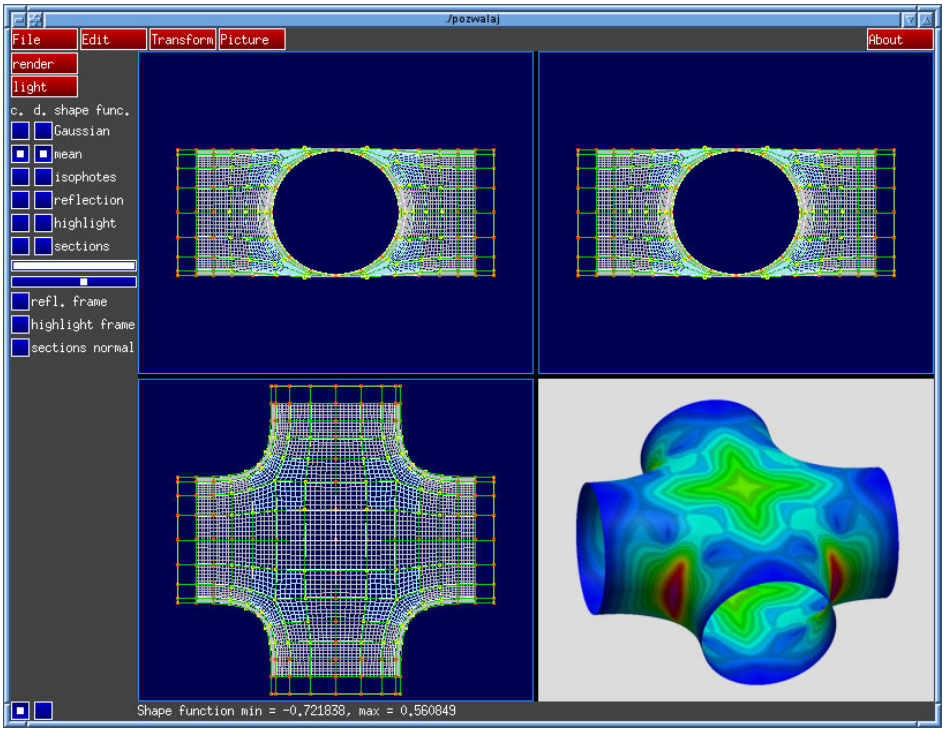
The marking of each vertex consists of five bits. They may be selected for manipulating by five switches just below the mark/unmark switch. We process two groups of vertices, so we need two bits. At first we mark the vertices shown below (the marked vertices are red):



Then we click the Transform button. There we have the coordinates of the reference point $[0,0,0]$ and reference vector $[-1,0,0]$, which determine the axis of the cylinder, and radius 1. Clicking the project on cylinder button makes the program project all the marked vertices on this cylinder. Then we click Edit again, and choose the second bit to mark/unmark (and we turn off switching the first bit). We mark the second set of vertices, then we click Transform, we enter the reference vector (direction of the cylinder axis) $[0,1,0]$ and again we project the vertices on the second cylinder. The result is as on the picture:

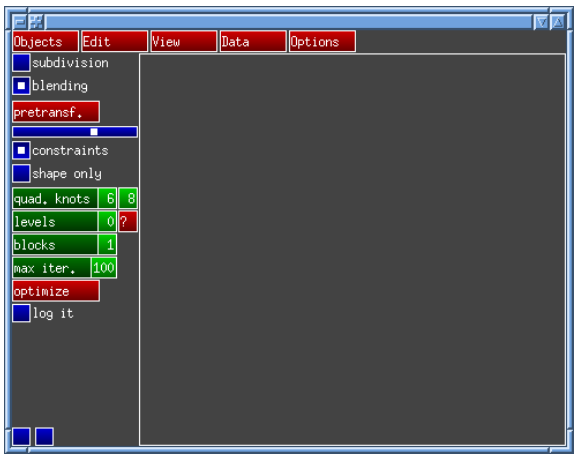


Now we click the Edit button in the second window and then refine. The refinement clears any vertex marking, therefore we go to the first window and in the similar way we mark the vertices and then we project them onto two cylinders, but this time we choose the cylinder radius 0.9267795297. After projecting the vertices we may inspect the surface. To do this, we click the View button in the second window, where we turn on displaying the surface and patches filling the holes in it. Then in the first window we click the button labelled Picture, then shape f., and we may choose the shape function to visualise. Clicking render starts the rendering process (which is ray tracing). An image of mean curvature obtained in this way is as follows:

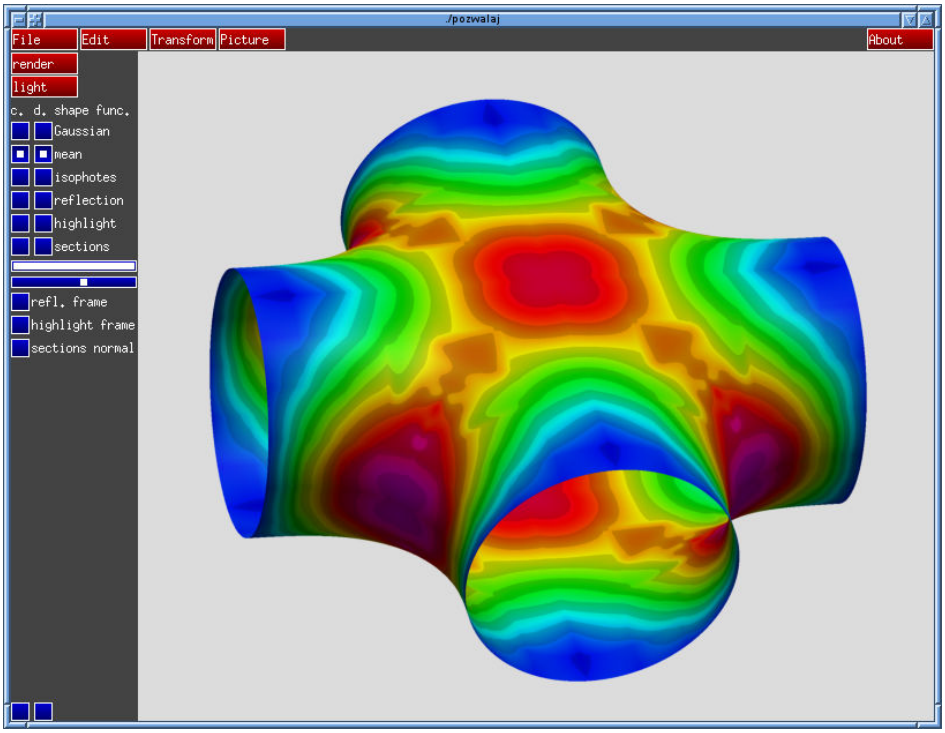


Time to optimise the surface shape. To do this, we need both marked bits to be selected in the editing menu, as the constraints, which we want, are imposed by fixing all vertices having one of the currently selected bits set. The boundary vertices (which we also marked in order to project them on the cylinders) are always fixed for the optimization procedure.

In the second window we click Options and then we turn on the blending switch. Then we turn on the switch labelled constraints. Here is, how the second window should look:



Now we click the optimize button. Some data are written out in the terminal, from which the program has been invoked. Intermediate results (after subsequent iterations of the optimization procedure) are displayed in the first window, and the user may still interact, e.g. in order to look at these results from different sides (this is useful during the optimization of surfaces represented by meshes with large numbers of vertices). The computations on a PC with 3.0GHz Intel Core 2 processor took less than 20 seconds, after which we may render the surface again. To obtain a bigger picture, before doing that we may move the cursor to the perspective image area and type S. The result is the following:



So far, it is the only existing documentation of the program `pozwalaj`. A detailed description of all widgets etc. is yet to be written, which is a tough job, especially as the program may (and will) continuously change. Any reader of this document is encouraged to contact the author (i.e. me), ask questions and give comments. This could really help.

Przemysław Kiciak