SPECIALIZED VISUALIZATION SYSTEMS FOR DIFFERENTIAL GAMES

V.L. Averbukh, S.S. Kumkov, V.S. Patsko,
O.A. Pykhteev, D.A. Yurtaev
Institute of Mathematics and Mechanics,
Ural Branch of Russian Academy of
Sciences, Ekaterinburg, Russia

3rd INTERNATIONAL IMACS MULTICONFERENCE on CIRCUITS, SYSTEMS, COMPUTERS and COMMUNICATIONS

4 – 8 July 1999

ATHENS, GREECE

The report is mainly devoted to computer systems for visualization of solutions of differential games of two classes. Both classes give three-dimensional objects for visualization.

The main object of solution of any differential game is its value function. So, to visualize the solution of a game means to visualize the value function of the game. The value function is usually calculated numerically, because its complete analitical computation can be made in very simple cases only.

Abstract

Visualization environments designed specially for viewing some important objects in optimal control theory and differential games are described.

The first system is developed for drawing level sets of value function in linear differential games with fixed terminal time.

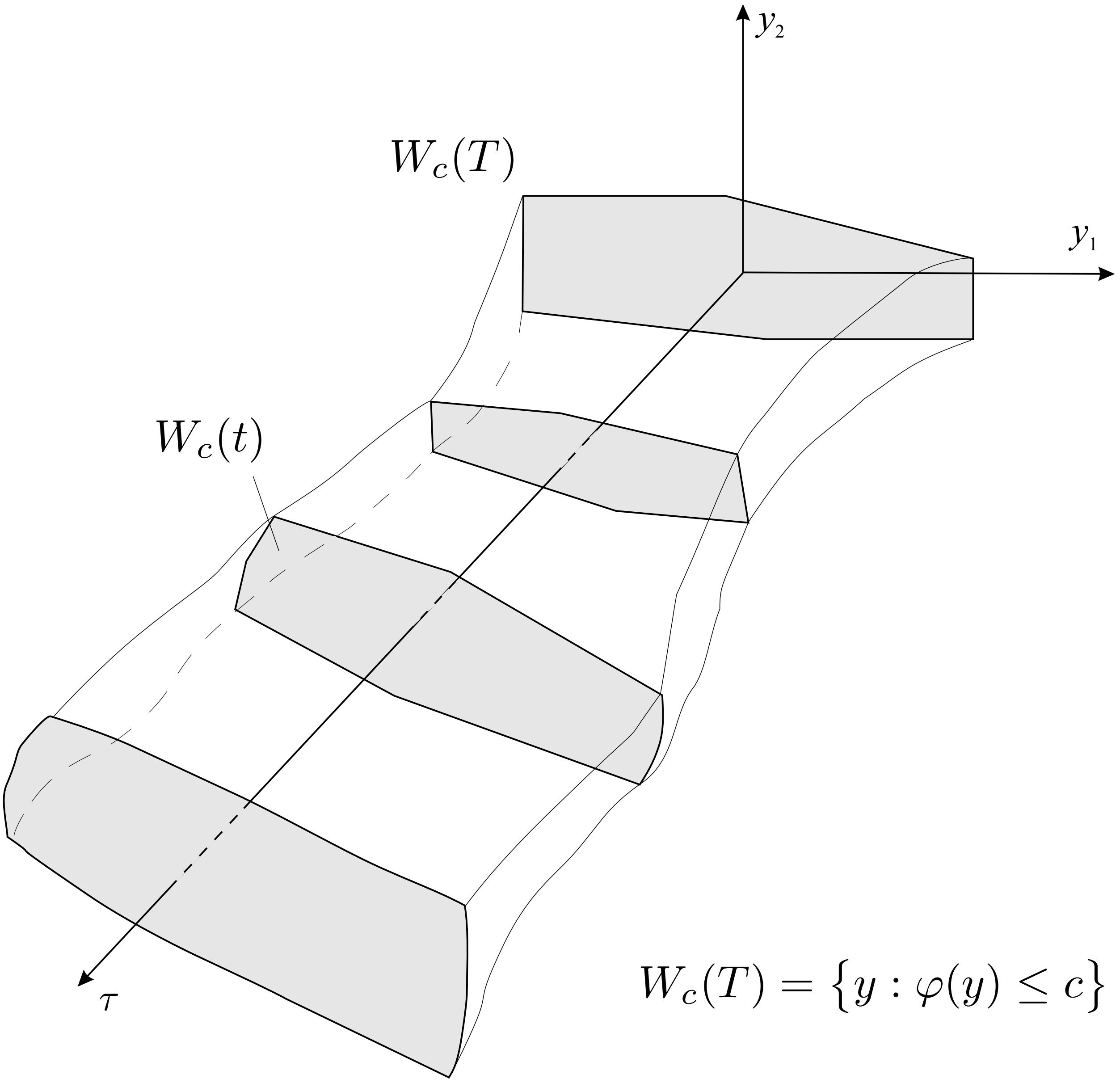
The second one is elaborated for the visualization of the value function graph in time-optimal problems.

Keywords: differential games, numerical simulation, computer 3D graphics, 3D objects manipulation

The first class consists of linear differential games with fixed terminal time. Dynamics of such games can be reduced to two-dimensional in the case when the terminal payoff function depends on two coordinates of the phase vector at the terminal instant. So, two reduced coordinates and the time give us three-dimentional game space.

The second class is two-dimensional pursuit-evasion games. Here, we have three-dimensional space of two phase coordinates and the meaning of the optimal result function.

LEVEL SET OF VALUE FUNCTION



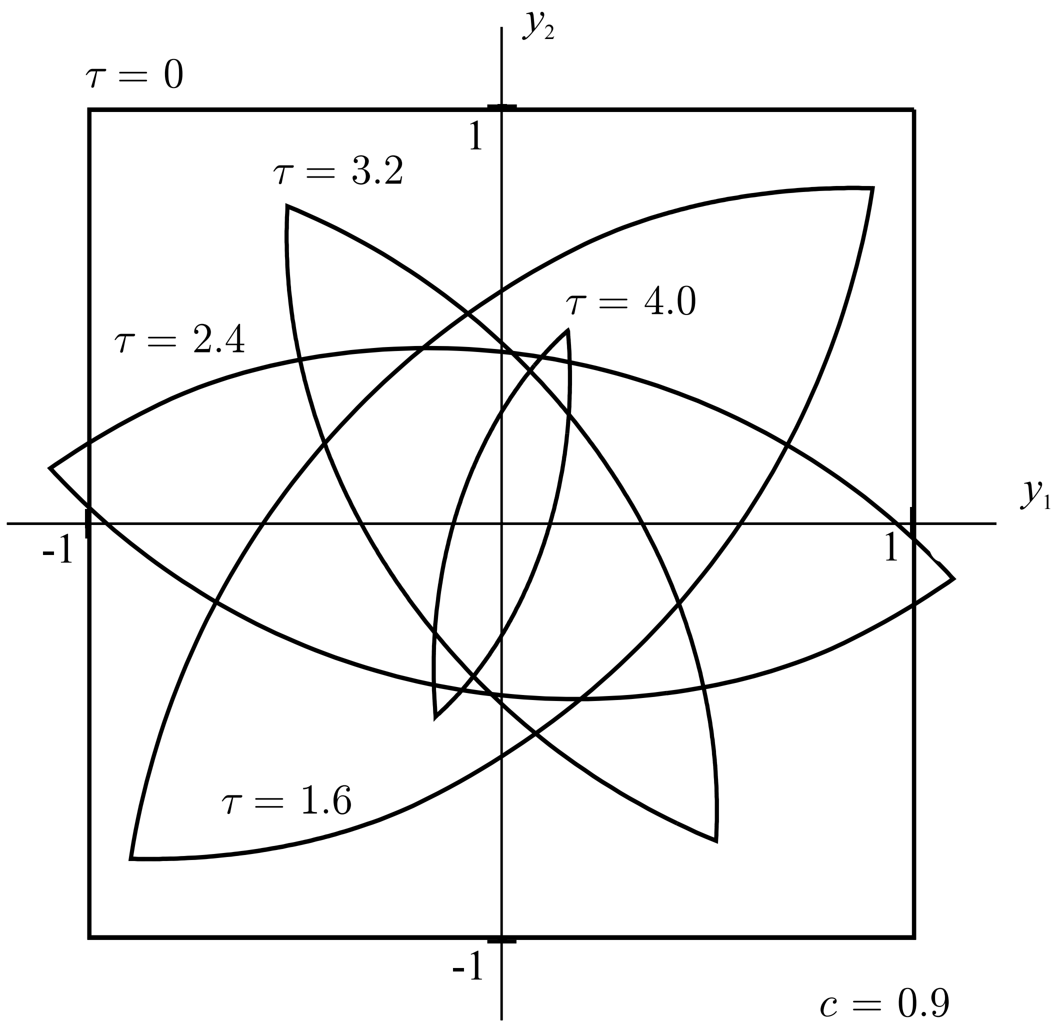
$$W_c(T) = \{y : \varphi(y) \leq c\}$$

For linear differential game with fixed terminal time and payoff function depending on two coordinates of the phase vector, there are algorithms allowing to calculate level sets of the value function. Each level set can be imaginated as a tube in three-dimensional space of the phase coordinates and time. Algorithms give these tubes as a set of polygonal sections orthogonal to the backward time axis.

SOME TIME SECTIONS OF LEVEL SET

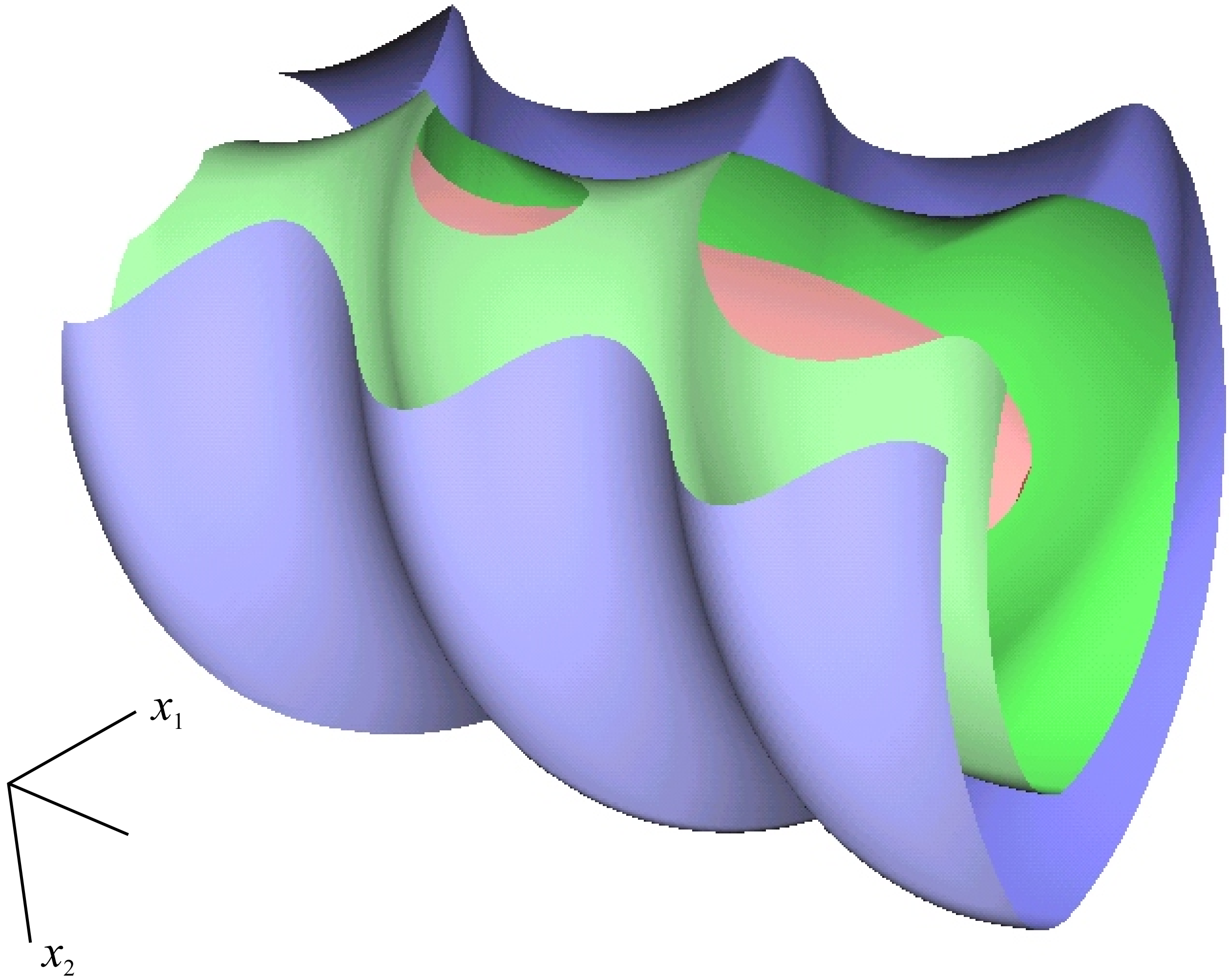
$$\dot{x}_1 = x_2 + v$$
$$\dot{x}_2 = -x_1 + u$$

$$|u| \leq 1,\ |v| \leq 1$$



These computational algorithms were developed in the middle of 80's. At that time, computers did not allow to make good visualization. So, the usual way to see a level set was to look at a number of its sections drawn in the plane of phase coordinates. Such pictures become uninformative with growth of shown section quantity.

VALUE FUNCTION LEVEL SETS FOR CONFLICT CONTROLLED OSCILLATOR



$\dot{x}_1 = x_2 + v$ $\quad t \in [0, 8]$

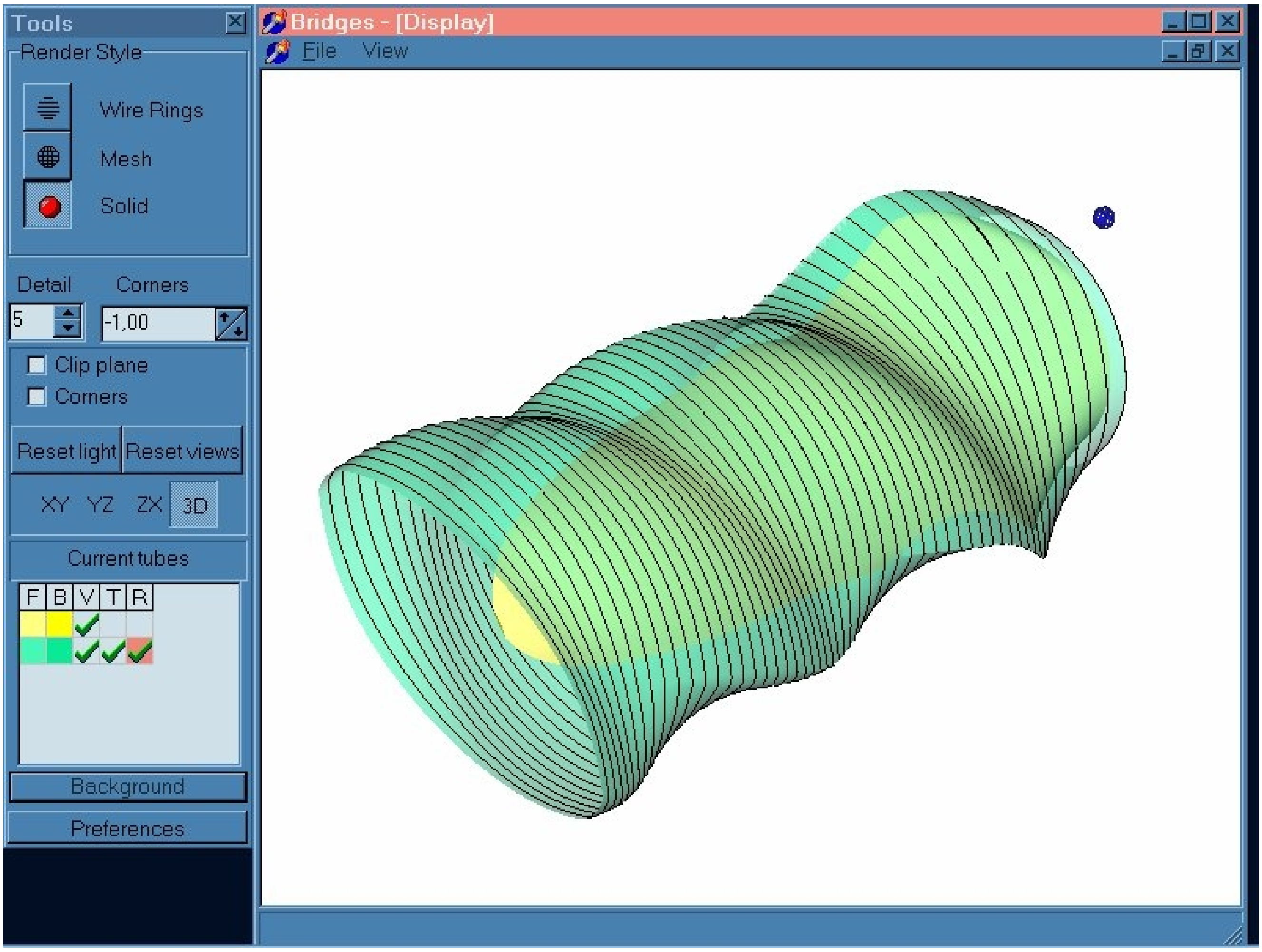
$\dot{x}_2 = -x_1 + u$ $\quad c = 1.05,\ 1.4,\ 2.7$

$|u| \leq 1,\ |v| \leq 0.9$ $\quad \varphi(x_1, x_2) = x_1^2 + x_2^2$ — payoff

IMM

Newly developed program allows to visualize a level set as a three-dimensional object. Also, it can draw a number of tubes simultaneously.

VISUALIZATION OF TUBES (Screenshot)



Detail: 5

Corners: -1,00

□ Clip plane

□ Corners

F	B	V	T	R
🟨	🟨	✓		
🟩	🟩	✓	✓	✓

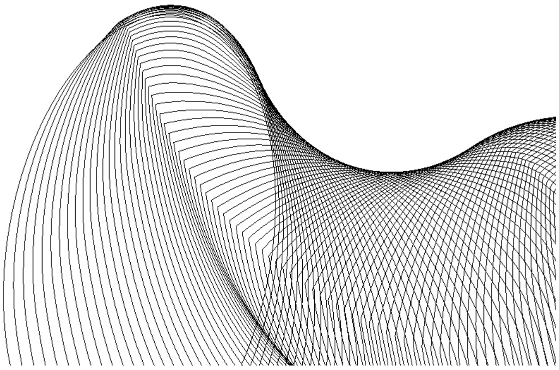
The following parameters can be changed via control panel: drawing regime (wire rings/mesh/solid), projection plane, number of objects, their color, transparency, section contours, presence of cutting plane.

A screenshot of the main window of the program is shown in this slide.

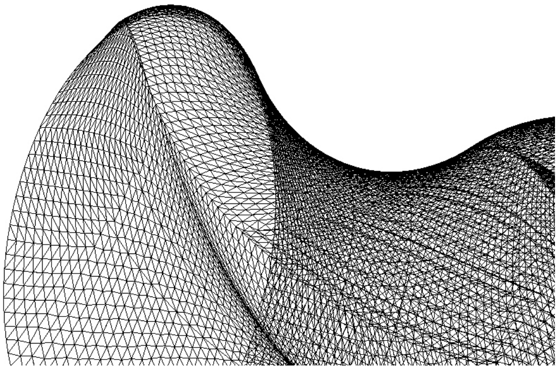
Originally, this program was developed for Unix operational system with Motif window manager. Further, it was ported to MS Windows 95/NT 4.0. The OpenGL language library was used for graphics.

In general, this program can show one or several level sets and allows to change their orientation in the space, color, opaqueness/transparency, position of light source, presence of contour skeleton, presence of cutting plane.

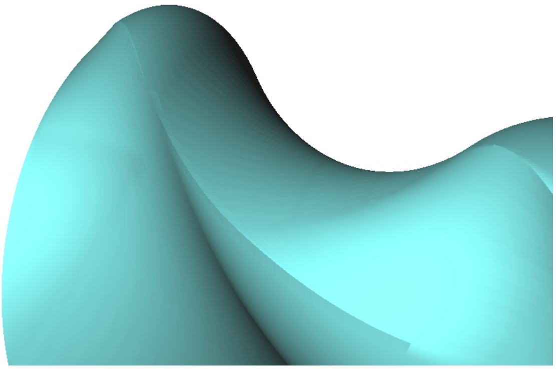
DEMONSTRATION OF VIEW REGIMES



Contour view



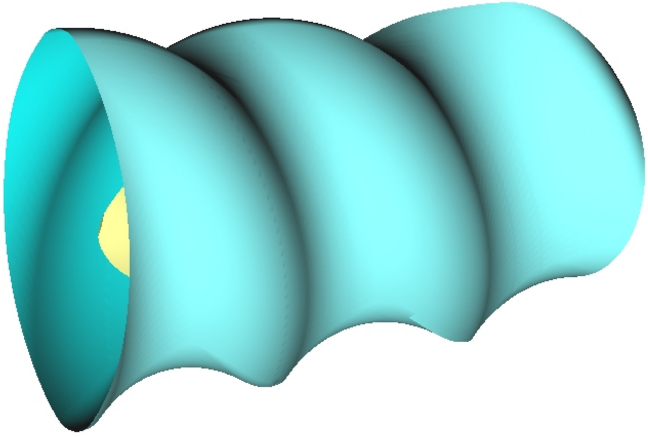
Mesh view



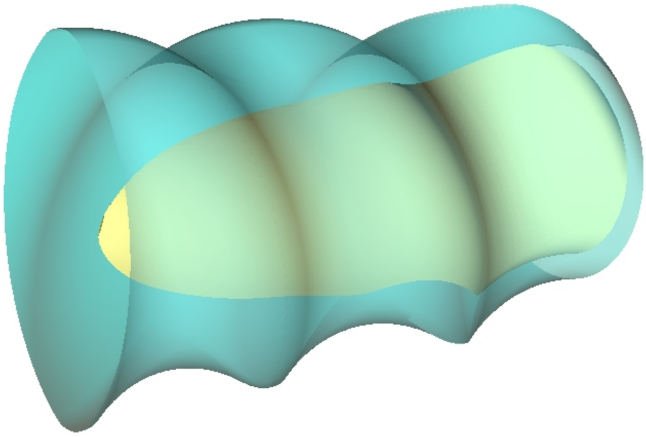
Solid view

Also, there are a number of drawing regimes: namely, the "wire ring" regime, the mesh regime and the solid one. The first regime is useful for primary orientation of the objects, because drawing a solid tube takes a time. When a proper view is found, the solid regime can be switched on. The mesh view was made mainly for debugging proposals: to control how the surface of a tube is reconstructed from separate sections.

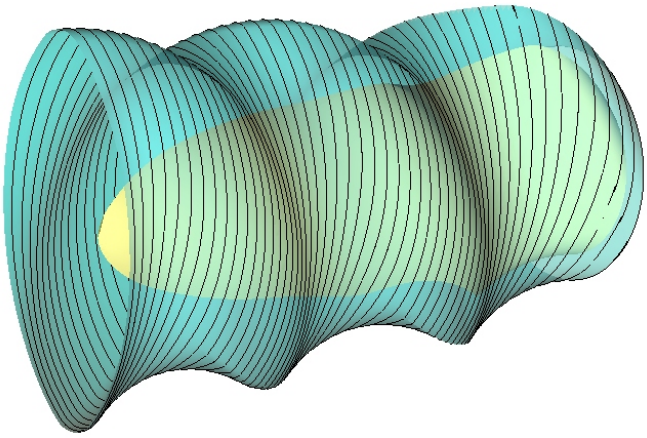
DEMONSTRATION OF VIEW REGIMES



Opaque view



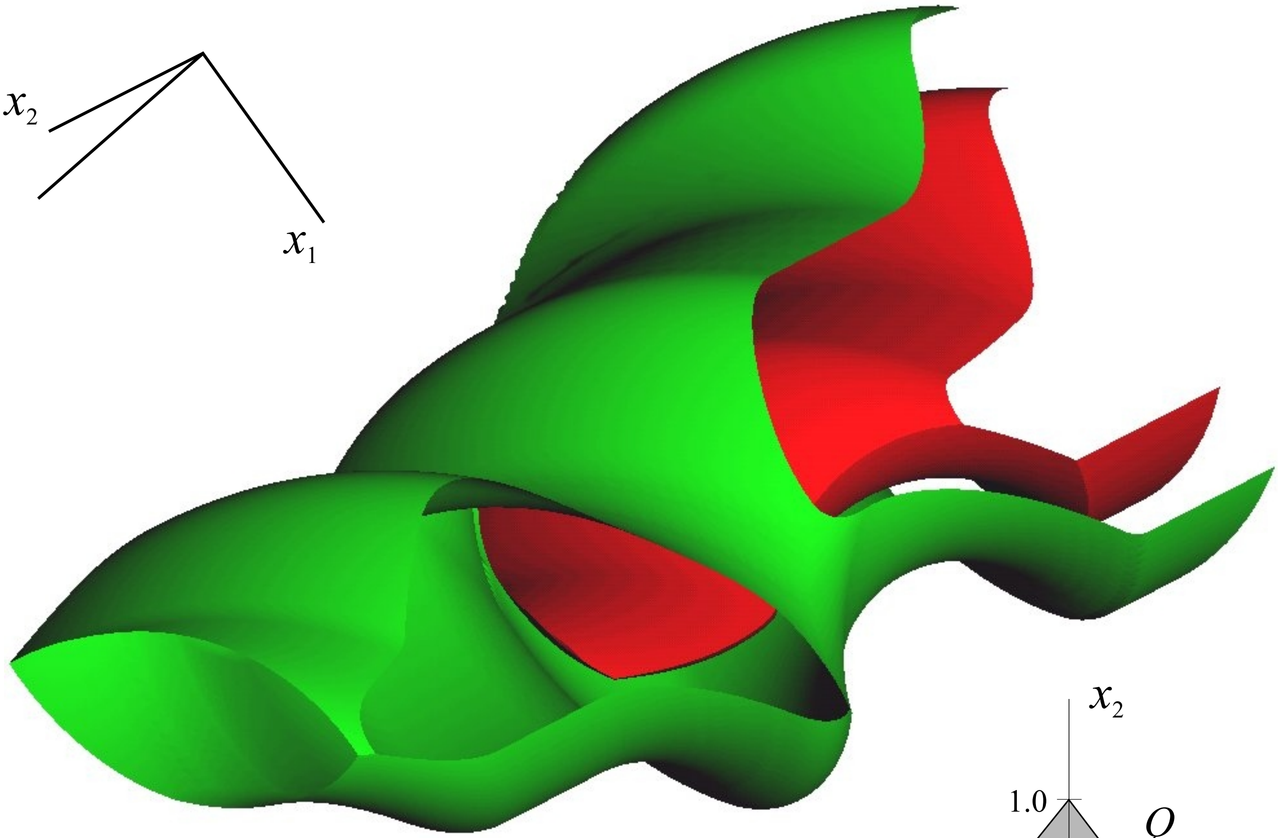
Transparent view

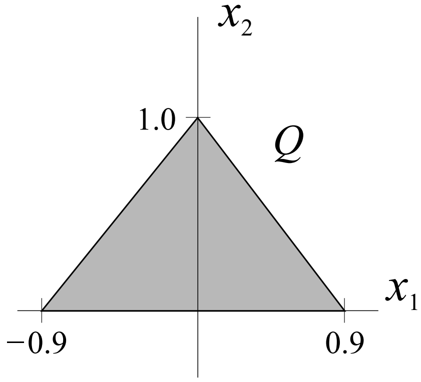


Transparent view
with contour skeleton

The program gives possibility to visualize several level sets simultaneously. The first way is to draw opaque tubes, but then the internal tubes are covered by external ones. To see the whole structure, the external tubes can be made transparent. But in this case, their three-dimensional structure can be lost. To restore it, the contour skeleton can be shown.

VALUE FUNCTION LEVEL SETS FOR CONFLICT CONTROLLED OSCILLATOR





$\dot{x}_1 = x_2 + v_1$ $\quad t \in [0, 8], \quad c = 1.4,\ 2.7$

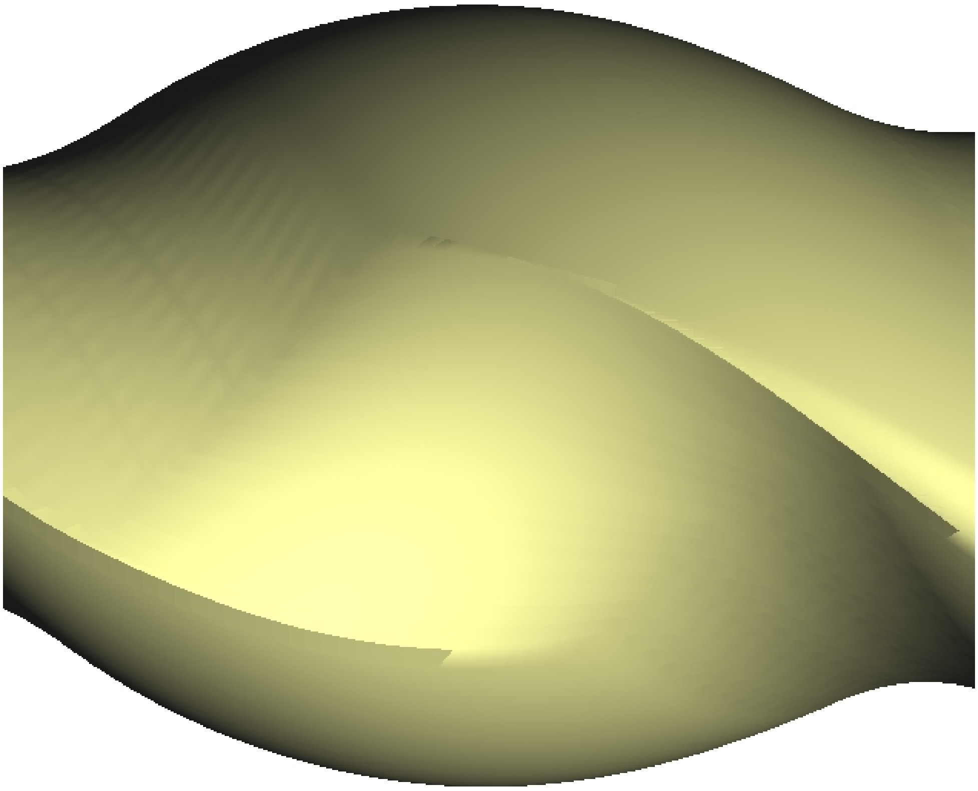
$\dot{x}_2 = -x_1 + u + v_2$ $\quad \varphi(x_1, x_2) = x_1^2 + x_2^2$

$|u| \le 1,\ v \in Q$ $\quad \varphi$ — terminal payoff function

IMM

Another way to explore a number of tubes is to cut them by a plane which position can be set by the user.

DISCONTINUOUS SINGULAR LINE

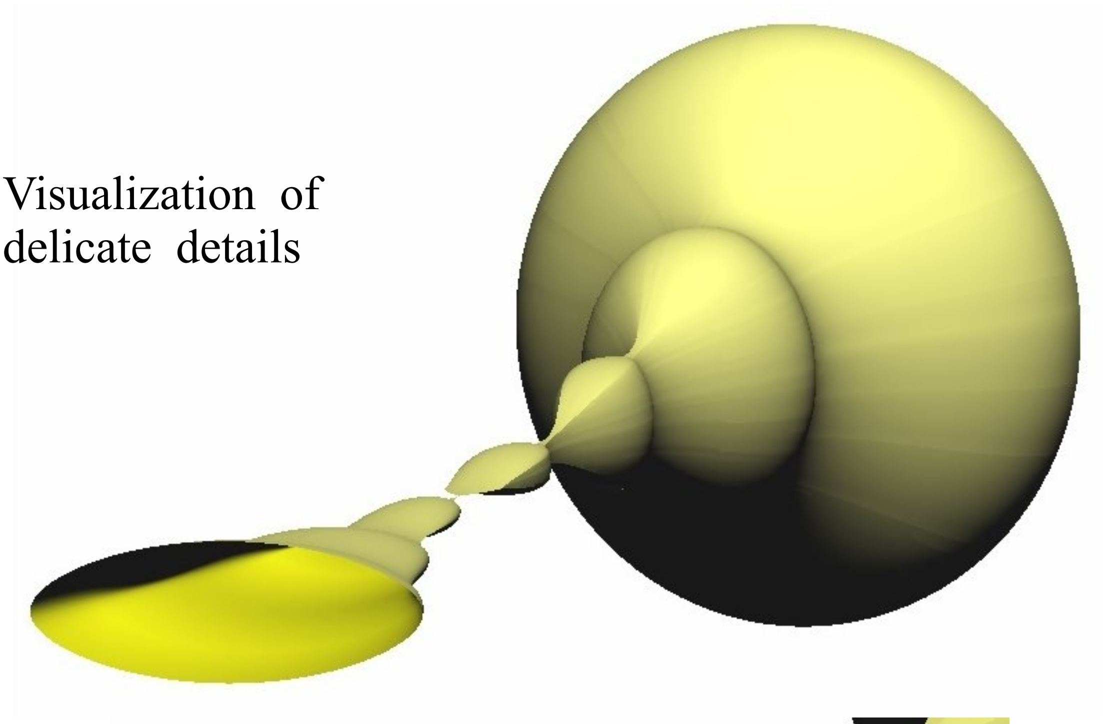


Oscillator system

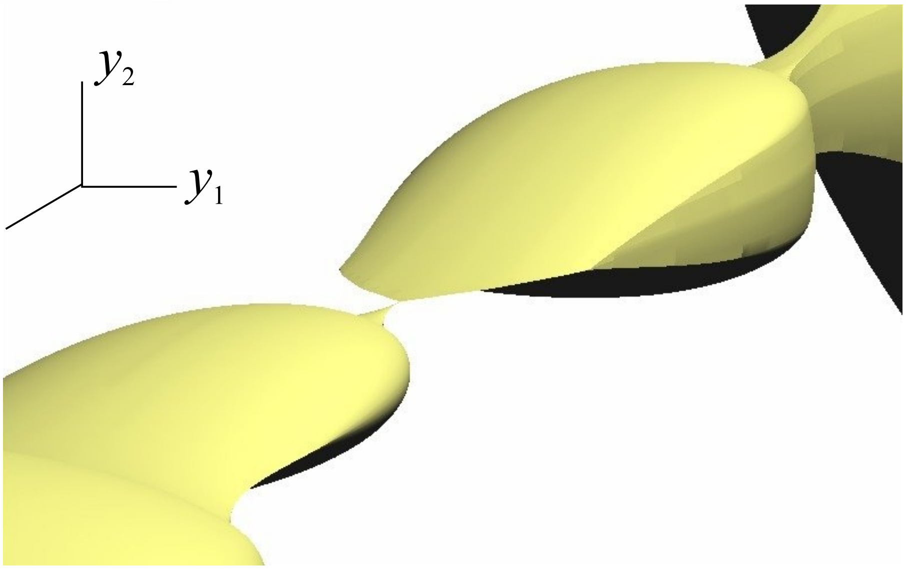
IMM

This program gives opportunity to investigate singularities of the value function. Usually, singularities appear as non-smoothnesses of the surface of a level set.

LEVEL SET WITH A “THROAT” (Pontryagin’s test example)

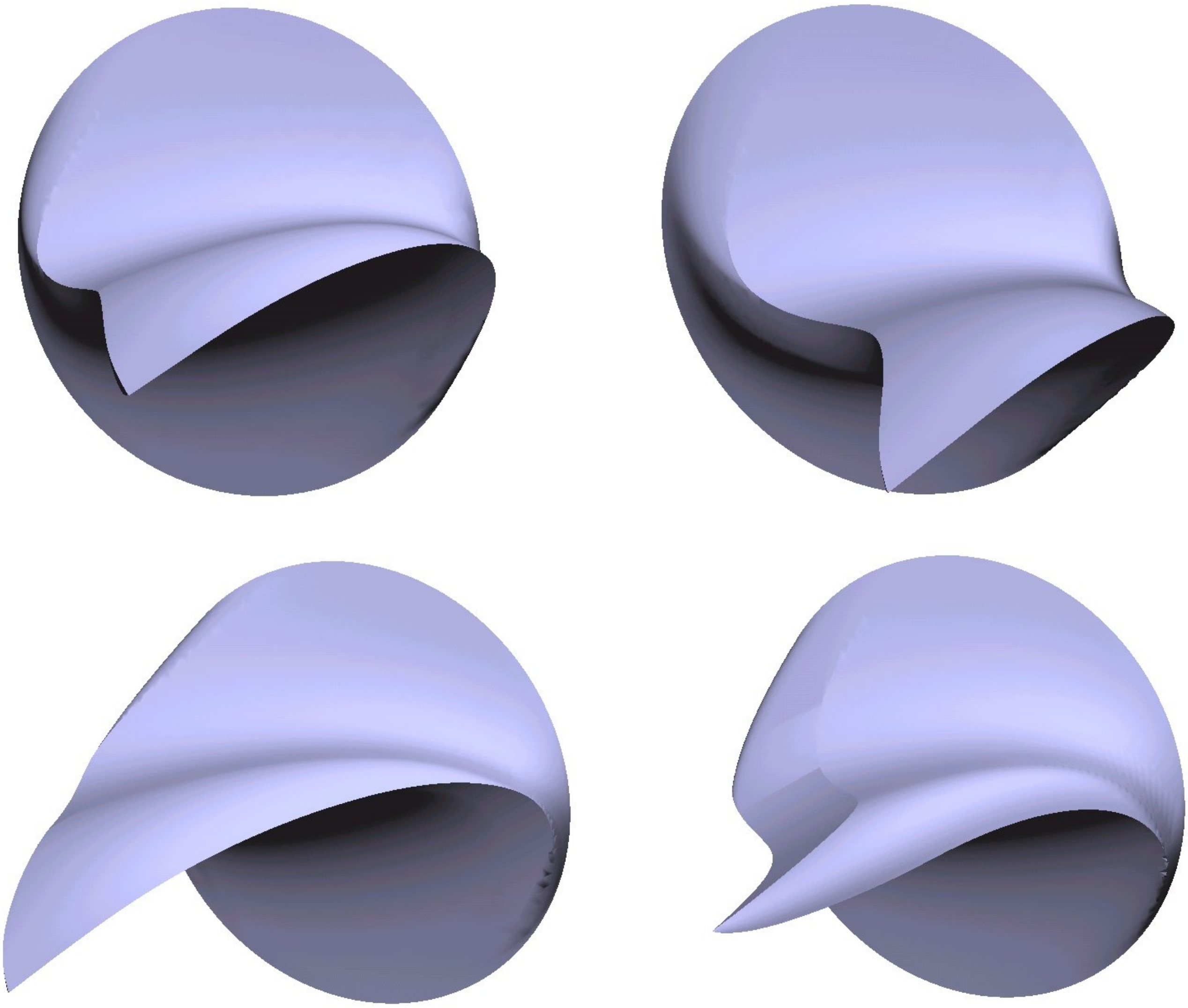


Visualization of delicate details



Also, it is possible to explore narrow "throats" of level sets by means of the program.

FOUR "DRAWING-PINS" (Evolution of the level set caused by changing the game parameters)



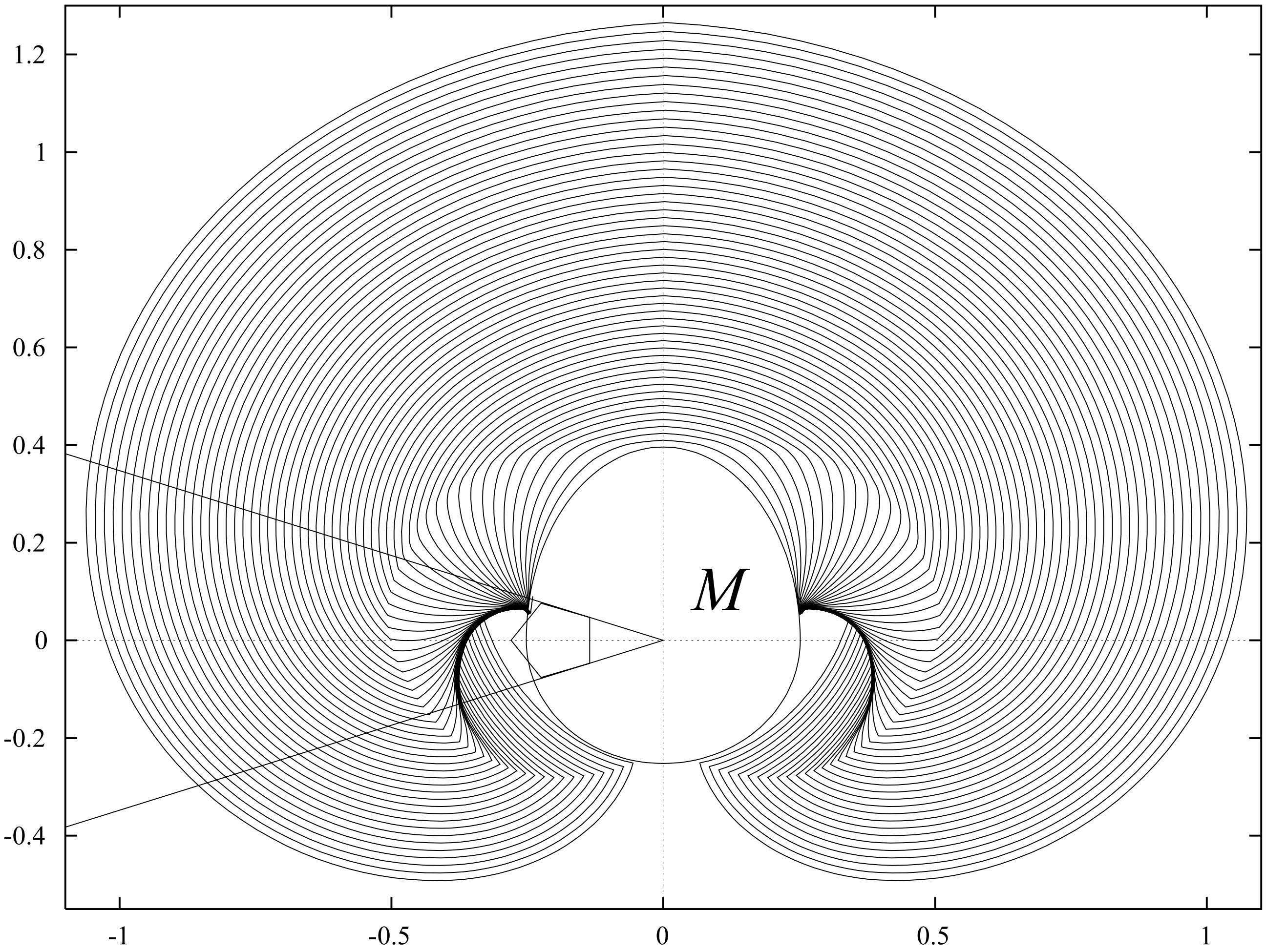
Similar sections – different bodies

Quick and accurate interactive visualization allows to see the dependence of level sets on parameters of the game.

The second problem is connected to visualization of the graph of the value function in time-optimal games of the second order on the phase variable. The payoff function is the time of approaching the terminal set. In the Institute of Mathematics and Mechanics, backward algorithms for constructing fronts of level sets of the value function are elaborated. Fronts are computed on a time grid. Each front is a polygonal line defined by its vertices. Neighbor fronts can have different number of connected components. Visualization program using the data on the fronts approximates the graph of the function and shows it. The graph is usually a complicated surface in three-dimensional space.

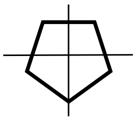
Below, there are the results of application of this program to two games with the "homicidal chauffeur" dynamics. The collections of the fronts for the following pictures were calculated by V.L.Turova.

TIME-OPTIMAL GAME PROBLEM
(Homicidal Chauffeur Game)



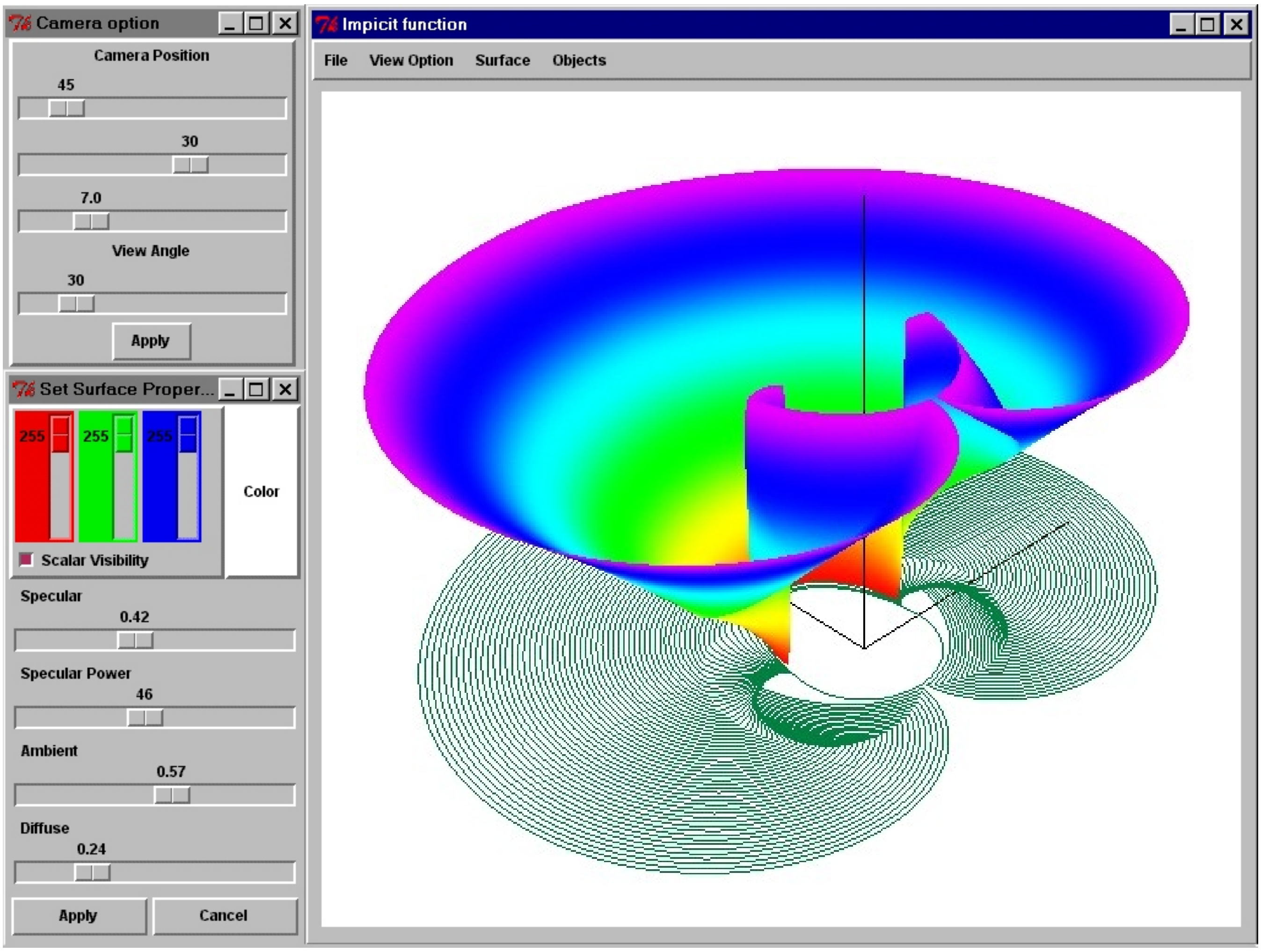
$$\dot{x}_1 = -\frac{w^{(1)}}{R} x_2 u + v_1$$

$$\dot{x}_2 = \frac{w^{(1)}}{R} x_1 u + v_2 - w^{(2)}$$

$|u| \le 1,\ \|v\| \in Q =$ 
$\operatorname{diam} Q = 0.8$
$w^{(1)} = 2,\ R = 2,$
Payoff is the time of approach M

Here, the fronts for the classic "homicidal chauffeur" problem are shown. The termial set denoted by M is egg-like. The fracture line of fronts can be seen. Also, areas of the front accumulation are evident. But from the picture, it is unclear whether the value function is conitunuous there or not.

VISUALIZATION OF OPTIMAL RESULT FUNCTION (Screenshot)



Camera Position

45

30

7.0

View Angle

30

255 255 255

■ Scalar Visibility

Specular 0.42

Specular Power 46

Ambient 0.57

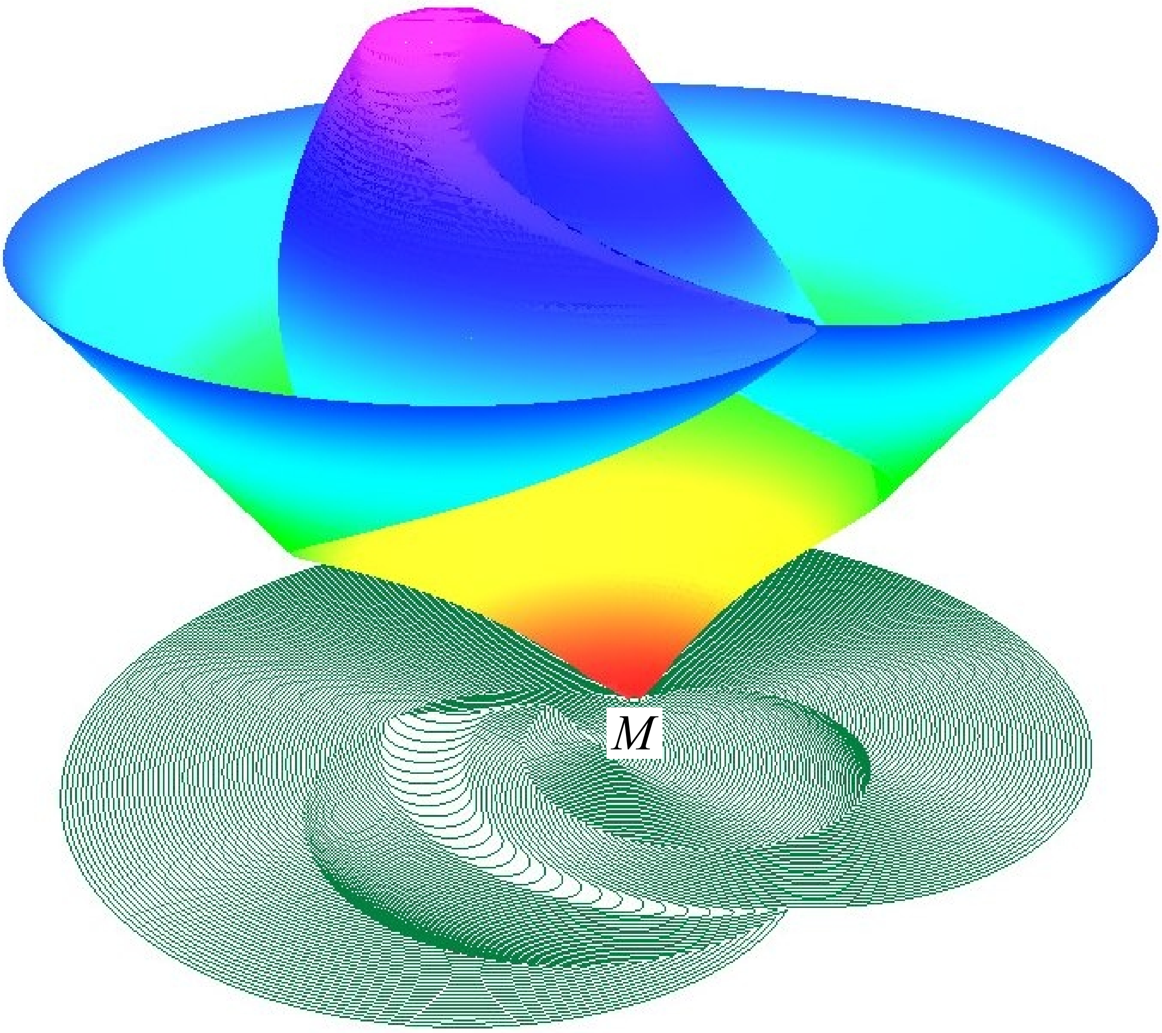
Diffuse 0.24

The surface is constructed by means of a separate program. Via control panel, the space orientation of the object and its coloring can be changed.
Both the surface and its level lines are shown.

This slide contains the screenshot of the main window of the program with three-dimensional picture corresponding to the fronts from the previous slide. Now, it is obvious that the value function is continuous.

The program has controls for changing the viewpoint, the direction of sight, the properties of the surface, the location of the radiant, etc. There is a painting regime when the surface color changes with the meaning of the drawing function. Fronts can be drawn in the horizontal plane.

“HOMICIDAL CHAUFFEUR” GAME. VALUE FUNCTION



$$\dot{x}_1 = -\frac{w^{(1)}}{R} x_2 u + v_1$$

$$\dot{x}_2 = \frac{w^{(1)}}{R} x_1 u + v_2 - w^{(2)}$$

$|u| \le 1, \ \|v\| \le 0.6$

$w^{(1)} = 2, \ R = 2,$

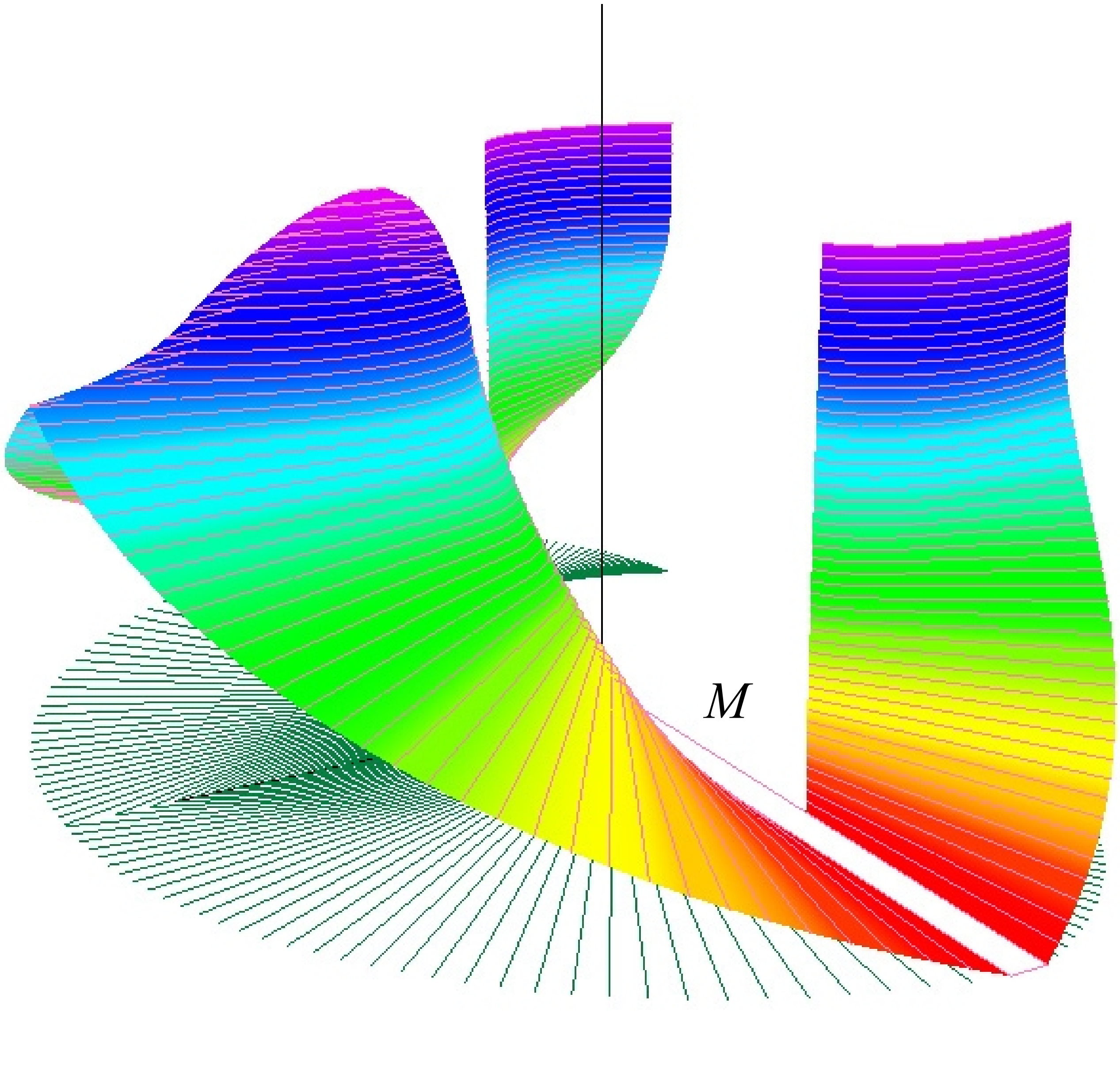
Payoff is the time of approach M

IMM

In this slide, a graph of the value function for the same game, but with the terminal set taken as a very small circle moved away the origin, is presented.

During the evolution, an instatnt appears when the front touches itself. After that, the fronts become disconnected and are divided into two parts: internal and external. In this picture, the internal area is fully filled. The external parts of the fronts are not shown after the touch instant. Here, the structure of the value function in the central area is of great interest.

VALUE FUNCTION FOR A VARIANT OF THE "HOMICIDAL CHAUFFEUR" GAME

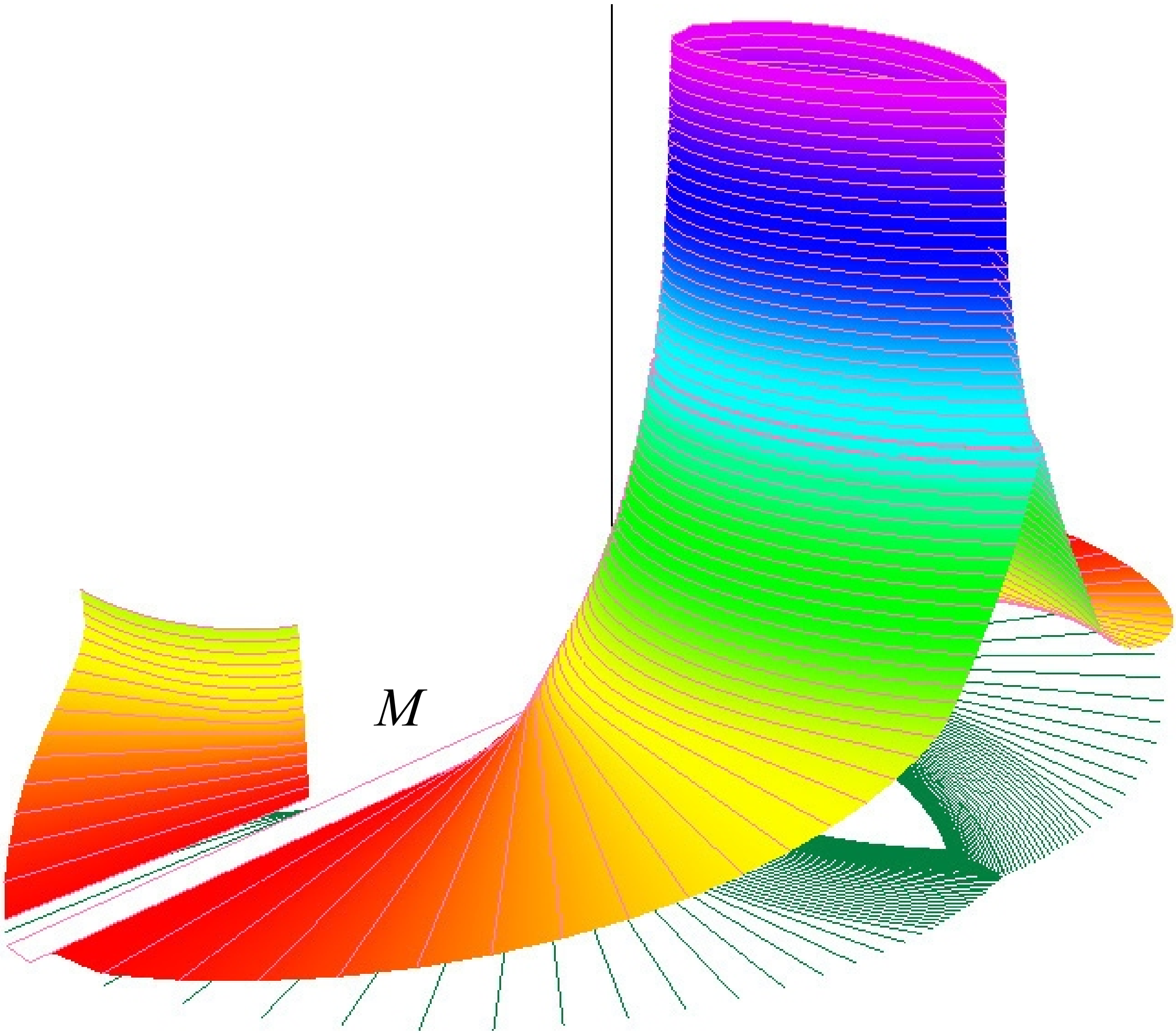


Here, the value function graph is shown for modified "homicidal chauffeur" problem suggested by P. Bernhard. In this problem, the constraints for the second player (evader) velocity depends on the distance to the first player (pursuer). In the reduced coordinates, the dependence is on the distance from the system position to the origin. The terminal set is a rectangle stretched along the first coordinate axis.

In this case, the value function is finite in front of the terminal set. Behind it, there are infinitely growing "petals".

On the surface of the graph, there are contours of level lines.

VALUE FUNCTION FOR A VARIANT OF THE "HOMICIDAL CHAUFFEUR" GAME



IMM

For the same dynamics, a bit different parameters of the second player constraints are taken.

Now in the front of the terminal set, a "smoke-stack" is appeared, inside which the value function is infinite.