2-D heat transfer problems in Scilab environment

Cite as: AIP Conference Proceedings **2172**, 090013 (2019); https://doi.org/10.1063/1.5133590
Published Online: 13 November 2019

Aleksey Nikolov, and Vesela Pasheva

2-D Heat Transfer Problems in Scilab Environment

Aleksey Nikolov[a)] and Vesela Pasheva[b)]

Department of Applied Mathematics and Informatics,
Technical University of Sofia, 1000 Sofia, Bulgaria

[a)]alekseyjnikolov@gmail.com
[b)]vvp@tu-sofia.bg

Abstract. Here we present our handy and convenient program for numerical solving of various heat transfer problems in 2-D regions composed of different material blocks. This application is based on Scilab.

Introduction

Due to the great practical importance of the heat transfer problems, there exist many computer programs that can be used for solving of such problems: ALGOR, FRAME, HEAT2, HEAT2R, HEAT3, ISOTHERM, MATLAB, RAMPANT, VOLTRA etc. Nevertheless, we also propose an application, which can deal with some two-dimensional heat transfer problems. This application is intended to be handy and easy to use and, at the same time, to have the most of important features in practice.

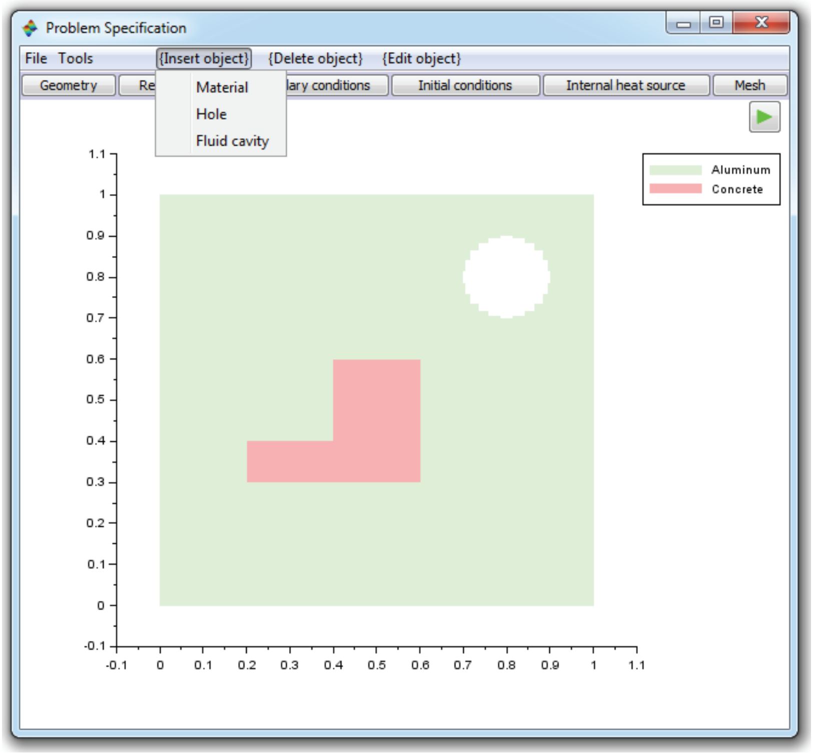


FIGURE 1. Inserting objects.

Our program is purposed to calculate the temperature distribution (and its corresponding flow field) in a planar region with a shape which may be composed of rectangular, triangular and elliptical (convex and concave) parts. This region represents a solid body consisting of different material block components, but also the body may contain areas with trapped fluid.

The computational algorithm as well as the graphic user interface are implemented on Scilab – this is a free software as well as a high-level programming language for scientific computing, one of the two major open-source alternatives of Matlab (the other one being GNU Octave). More information about modeling and simulation with Scilab can be found for example in [3].

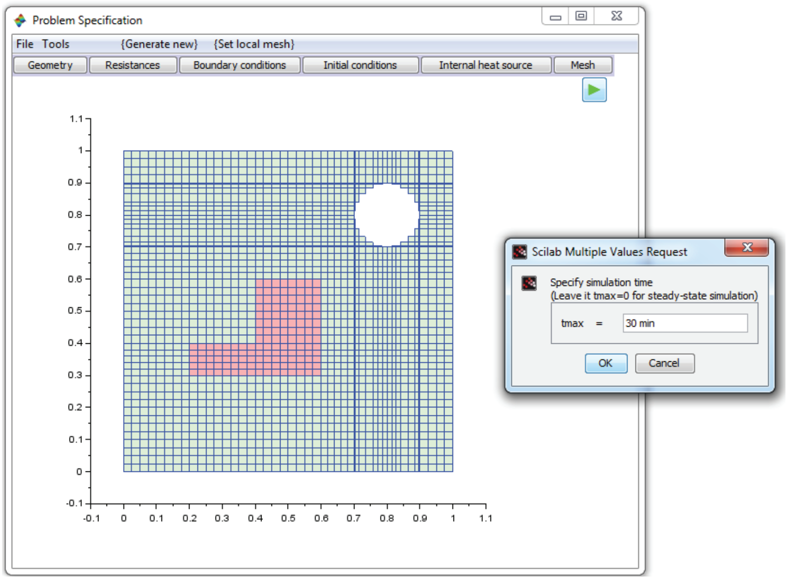
Typically, the heat conduction process is studied by use of the famous partial differential equation:

$$\frac{\partial}{\partial x}\left(\lambda \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\lambda \frac{\partial T}{\partial y}\right) + I(x, y, t) = C\,\frac{\partial T}{\partial t}, \tag{1}$$

where:

- $T(x, y, t)$ is the unknown temperature distribution in the body at a time t;
- λ gives the thermal conductivities at the points of the body;
- C gives their volumetric capacities;
- the function $I(x, y, t)$ represents the rate of internal heat generation.

In our case λ is a piecewise constant function (implying discontinuities) and then we may consider our solutions as generalized solutions of (1). Usually it is suitable to solve the problems for equation (1) by numerical methods: most often these are finite-difference methods or finite element method (see for example [1, 2, 4]). In our application we use an explicit finite-difference scheme, proposed by Blomberg [1], which is very apposite according to the considered problem's specificity and the Scilab's capabilities. However we still have the disadvantage to use only rectangular grids, implying that the sloped and the curved edges of the boundary should be approximated by zigzag boundary segments.



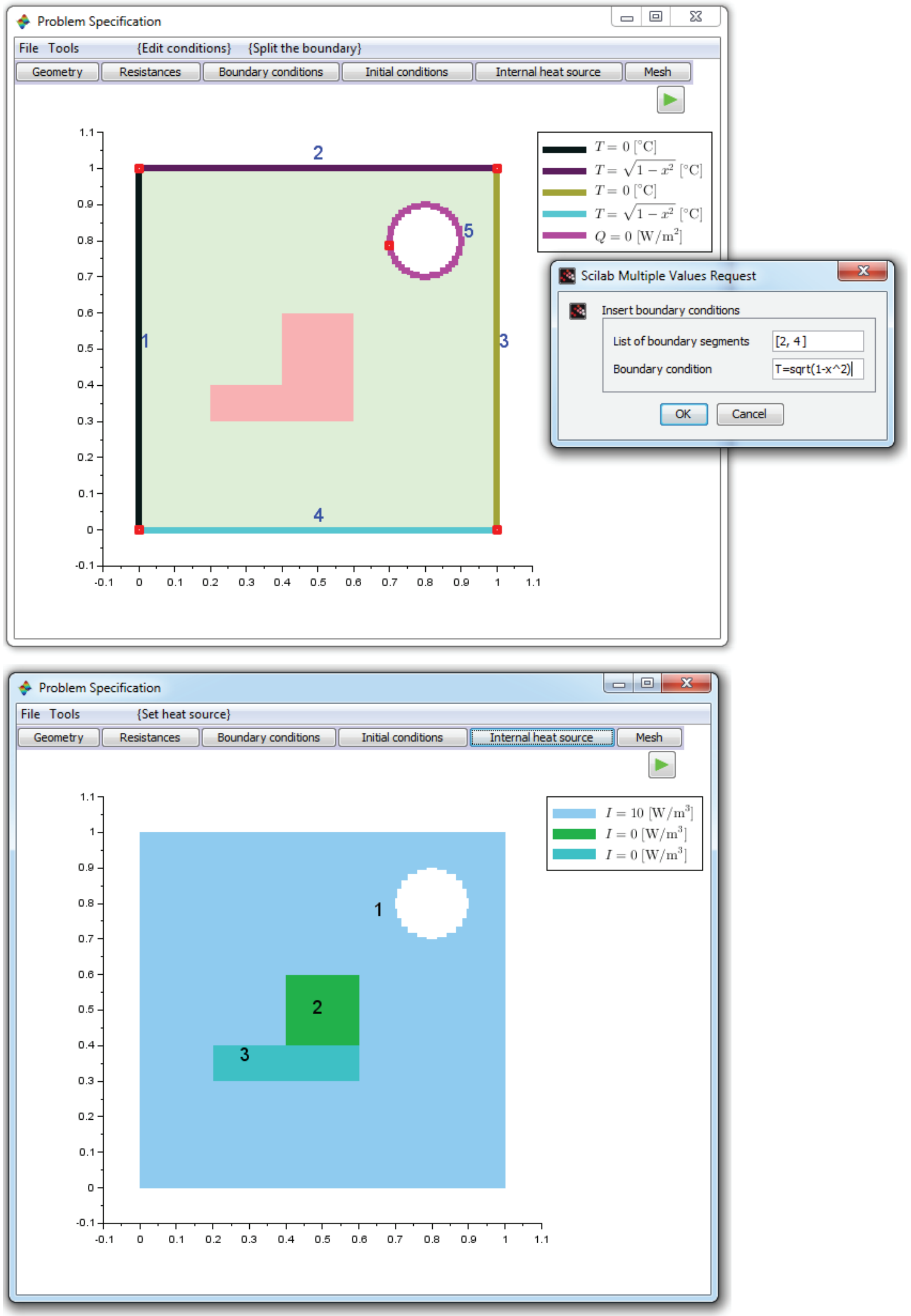
Specify simulation time
(Leave it tmax=0 for steady-state simulation)

tmax = 30 min

FIGURE 2. The numerical mesh and specifying the simulation time.

In the case when a trapped air is considered the mathematical model is definitely complicated, because the heat conduction is combined with long-wave radiation exchange described by nonlinear equations ([5]). The radiation depends on the fourth power of the cavity surface temperatures (in Kelvin). After the discretization, in our numerical solution at each time step a system of nonlinear equations should be solved by successive approximations method ([1]), which tangibly increases the time necessary for simulation.

Besides the radiation coupled to the heat conduction, a ventilation exchange between the air cavity and the ambient can be added to this model.


Problem Specification
File Tools {Edit conditions} {Split the boundary}
Geometry Resistances Boundary conditions Initial conditions Internal heat source Mesh
1
2
3
4
5
$T = 0$ [°C]
$T = \sqrt{1-x^2}$ [°C]
$T = 0$ [°C]
$T = \sqrt{1-x^2}$ [°C]
$Q = 0$ [W/m^2]
Scilab Multiple Values Request
OK
Cancel
Problem Specification
File Tools {Set heat source}
Geometry Resistances Boundary conditions Initial conditions Internal heat source Mesh
1
2
3
$I = 10$ [W/m^3]
$I = 0$ [W/m^3]
$I = 0$ [W/m^3]

Insert boundary conditions

List of boundary segments	[2, 4]
Boundary condition	T=sqrt(1-x^2)

FIGURE 3. Setting the boundary conditions and an internal heat source.

Basic Features of the Program

There are two main types of problems which can be solved by our program:

- **Steady-state problems.** Solving boundary value problems for equation (1) with zero right-hand side (in which case the equation becomes elliptic), to find the temperature distribution in the body when an equilibrium state is achieved.
- **Transient-state problems.** Solving the parabolic equation (1) with prescribed initial and boundary conditions, to find the temperature distribution in the body at a user specified moment of time.

Further, there are two types of boundary conditions which can be applied to each of the boundary segments:

- *Dirichlet type* – there are prescribed temperatures on the boundary (in $^\circ$C).
- *Neumann type* – there is prescribed heat flow (in $\mathrm{W/m^2}$) through the boundary into the region, where the heat flow is perpendicular to the boundary.

The user can specify the following input data into the Problem Specification window:

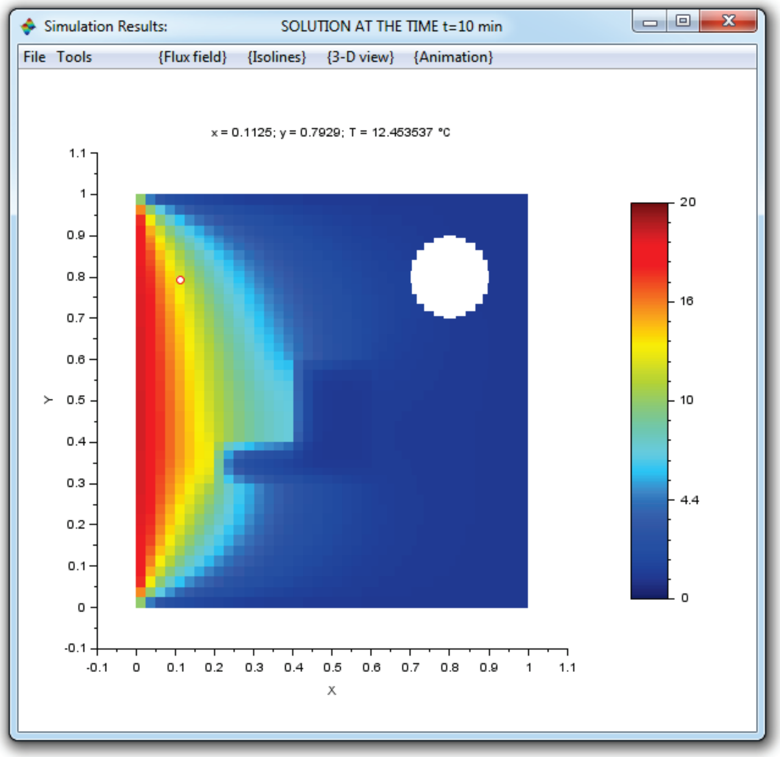
- **Geometry.** In order to compose the body, the user may insert objects (blocks) of different shapes by pup-up menu. The currently available shapes are: rectangles, right-angled triangles, ellipses, semi-ellipses and quarter-ellipses. The objects may overlap. In a place where two objects overlap the body takes the thermal properties of the one which is lying above. Each block should be assigned with some material picked from a list (a block can also be specified as a "hole", i.e. it deletes the lying below material). If the block represents a trapped air in a cavity, the user may set this cavity to be ventilated, specifying the corresponding ventilation rate and the inlet air temperature.
- **Resistances.** There is possibility to set contact surface resistances (in $\mathrm{m^2\,K/W}$) between the blocks inside the body as well as on the boundary segments of the body. From mathematical point of view it makes no sense to set resistances on boundary segments with Neumann boundary conditions, since these resistances would not be involved in the calculations.
- **Boundary conditions.** Once the region is created, the program divides the boundary into separate segments as the vertexes on the edges determine the split points. The user may add other split points, customizing in that way the boundary division. On each of the boundary segments the user may specify a different Dirichlet or Neumann boundary condition which can be a constant or, more commonly, a function of the Cartesian coordinates (x, y). Additionally, if the problem is not steady-state, this function may also depend on time t. It can be any available in Scilab function and it should be entered using the Scilab's syntax (the syntax of the basic functions in Scilab is the same as in Matlab). The expressions of the introduced functions are converted into LaTeX strings and correspondingly they are displayed on the figure as good looking formulas.
- **Initial conditions.** On each of the defined blocks (provided the block is not a "hole") an initial temperature distribution is prescribed by a function of (x, y). Of course, in the steady-state problems it is not necessary to specify the initial conditions.
- **Internal heat source.** A heat source function (analogously to the initial temperature distribution) can be specified in each block separately. The rate of internal heat generation (in $\mathrm{W/m^3}$) may be a function of (x, y, t), in the steady-state case only of (x, y).
- **Mesh.** The user controls the density of the generated numerical mesh setting the maximum step sizes in respect to the x-direction and to the y-direction. If it is necessary there is additional feature to generate more refined mesh in a chosen subregion.

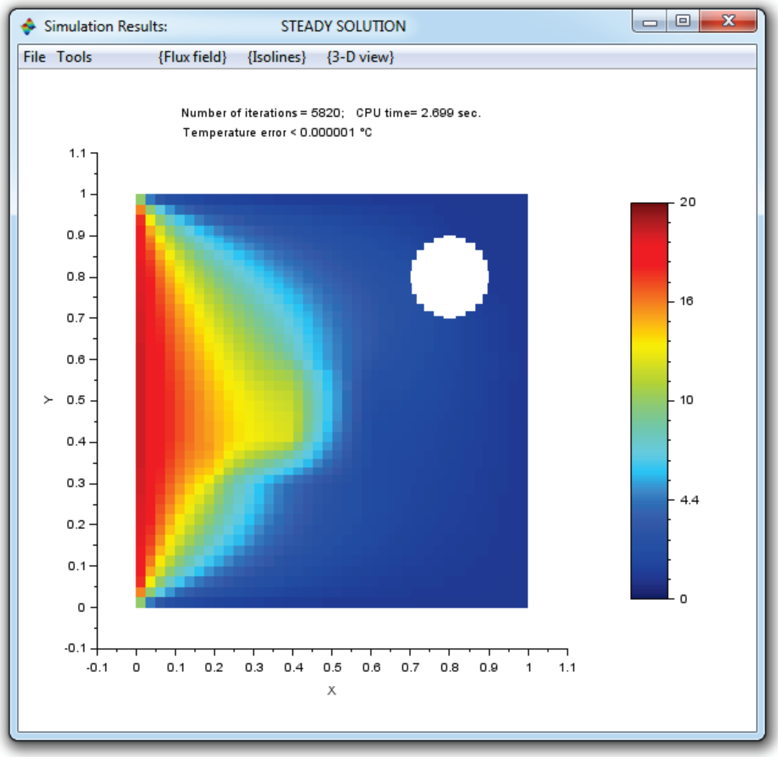
Output Data

The simulation results are structured in matrixes as well as they have proper visualization.

According to the used numerical scheme, the main output data is: a matrix T containing the output temperatures at the centers of the mesh cells, a matrix QX containing the calculated heat flows in the x-direction passing through the vertical edges of the mesh cells and a matrix QY containing the calculated heat flows in the y-direction passing through the horizontal edges of the mesh cells.

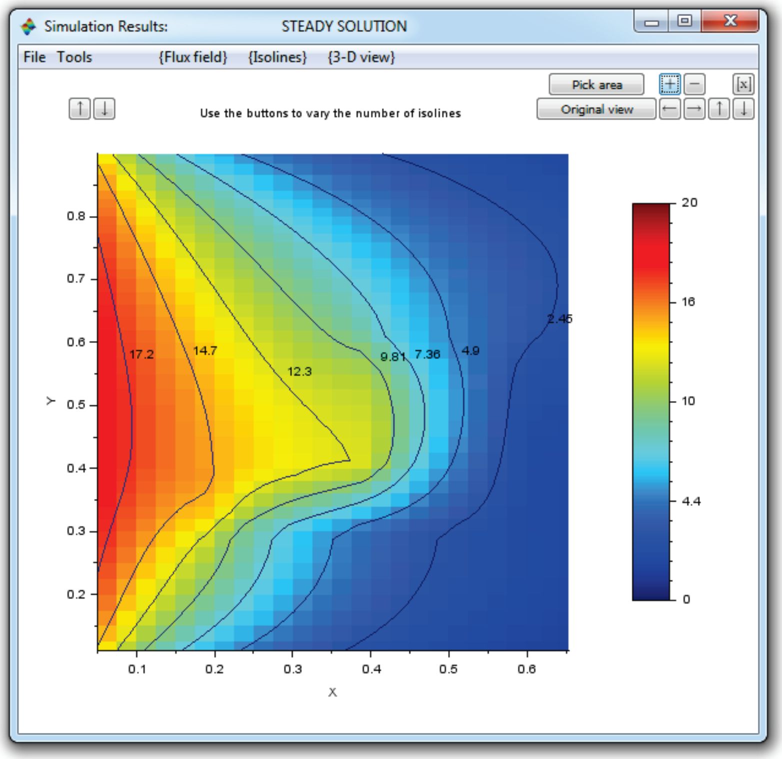
Different visualizations in the Simulation Results window are shown on Fig. 4-6 (2-D color map, isotherms, vector field, 3-D view). In the case of a transient-state problem the heat transfer process can be shown as animation.


Simulation Results: SOLUTION AT THE TIME t=10 min
File Tools {Flux field} {Isolines} {3-D view} {Animation}
x = 0.1125; y = 0.7929; T = 12.453537 °C
Y
X
20
16
10
4.4
0


Simulation Results: STEADY SOLUTION
File Tools {Flux field} {Isolines} {3-D view}
Y
X
20
16
10
4.4
0

Number of iterations = 5820; CPU time= 2.699 sec.
Temperature error < 0.000001 °C

FIGURE 4. Transient-state problem and steady-state problem solutions (temperature distributions).



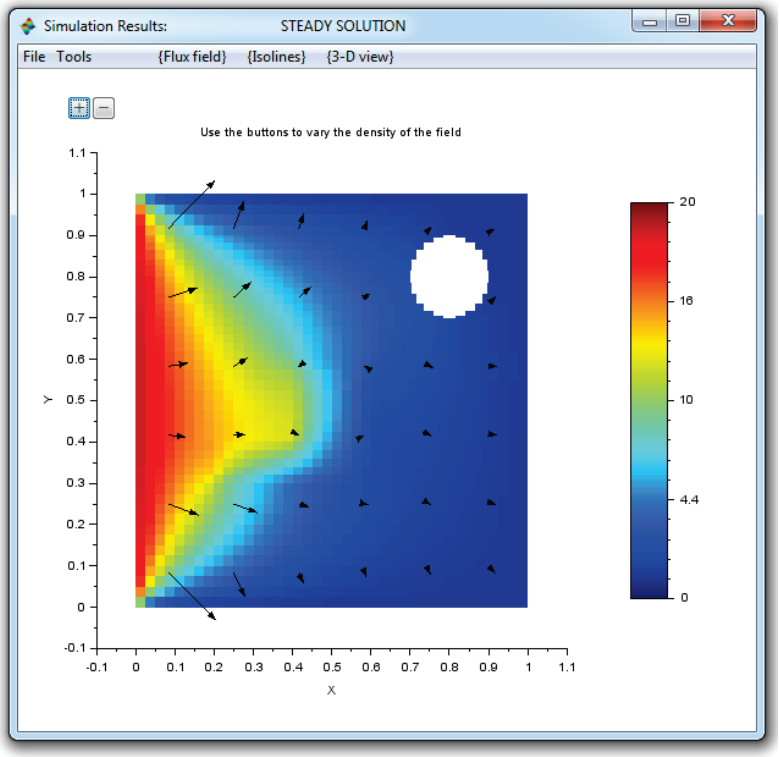

Simulation Results:
STEADY SOLUTION
File Tools
{Flux field} {Isolines} {3-D view}
Use the buttons to vary the density of the field
20
16
10
4.4
0
Y
X

FIGURE 5. Isotherms and vector field of the heat flux.

Acknowledgments

The work of Aleksey Nikolov was supported by the Bulgarian Ministry of Education and Science under the National Program for Research "Young Scientists and Postdoctoral Students". This program was approved with RMS No. 577/ 17.08.2018.

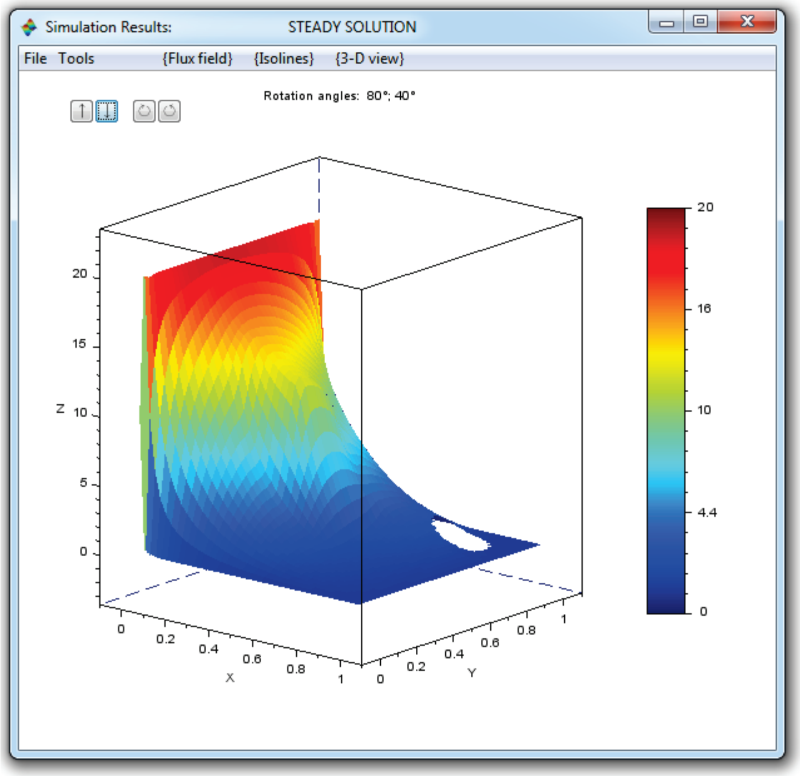


FIGURE 6. 3-D view of the solution.

REFERENCES

[1] T. Blomberg, *Heat conduction in two and three dimensions : computer modelling of building physics applications,* Byggnadsfysik LTH, Lunds Tekniska Högskola, (1996).
[2] S.C. Brenner and L.R. Scott, *The Mathematical Theory of Finite Elements Methods,* Texts in Applied Mathematics, vol. 15, Springer-Verlag New York, (2008).
[3] S.L. Campbell, J. Chancelier and R. Nikoukhah, *Modeling and Simulation in Scilab/Scicos,* Springer, (2006).
[4] C. Hirsch, *Numerical computation of internal and external flows,* John Wiley & Sons Ltd., (1992).
[5] R. Siegel and J.R. Howell, *Thermal radiation heat transfer,* Hemisphere Publishing Corporation, Washington DC, USA, (1992).