

CFD graphical interface in MATLAB

T. M. Pinho
School of Engineering
University of Minho
4800-058 Guimarães, Portugal

(Tiago.pinho@dps.uminho.pt)

S.F.C.F. Teixeira
School of Engineering
University of Minho
4800-058 Guimarães, Portugal

(st@dps.uminho.pt)

Abstract - The use of computational methods for solving partial differential equations in the field of Heat Transfer is crucial. However, the actual user friendly commercial applications, for the resolution of this kind of problems are expensive.

The purpose of this work is the development of a graphical interface for the resolution of heat transfer problems using the finite volume method. This development is made in MATLAB, due to its capacity to generate graphical results in an easy and efficient way.

This graphical interface is used as a learning tool in post graduation studies at University of Minho.

Keywords: CFD; MATLAB; graphical interface; educational software.

1.0 Introduction

Engineering problems are usually described by sets of partial differential equations. They are mathematical approximations of the physical reality and they can be solved by complex analytic methods but they often are solved using numerical algorithms implemented in computer programs.

There are some commercial software used to solve problems involving heat transfer and fluid dynamics phenomena (CFD). These commercial softwares are complex and they always solve standard cases. For a particular problem, it can sometimes be difficult to introduce particular aspects and special conditions. MATLAB and FLUENT are just two examples of this commercial software that have been used with success.

Teaching the solution of CFD problems at post graduation studies is always a challenge and using the commercial software can be expensive.

Following some previous work at the research group ([1], [2], [3]), a new CFD software was developed into a graphical interface in order to make the subject more attractive and allowing the user to define his problem. The interface was implemented in MATLAB environment because of its availability at the university campus and its graphical facilities.

2.0 MATLAB Environment

MATLAB is a commercial code to solve a wide variety of problems (Figure 1). This software has a friendly interface and works with a wide variety of *Toolbox* that can perform different type of problems: statistics, optimization, economics, signal processing.

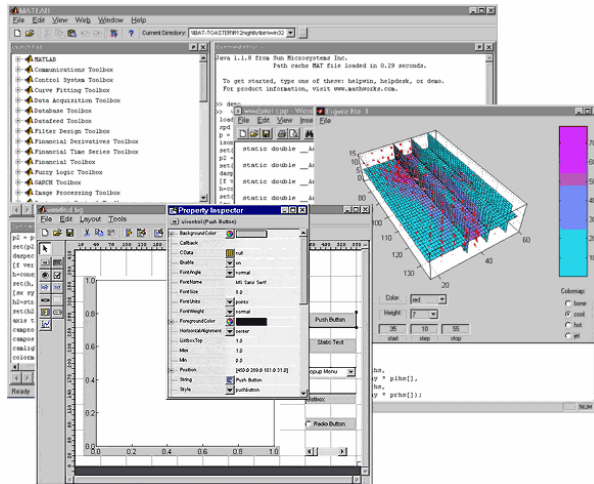


Figure 1 - Matlab work environment.

MATLAB has a specific Toolbox to solve partial differential equations, called PDE Toolbox. This software uses the finite element method to integrate the equations. More information on MATLAB and Toolboxes can be found at www.mathworks.com.

MATLAB also has different tools to create Graphical User Interface (GUI). These tools simplify the design and implementation processes and so they can be used to create new interfaces [4]. The Layout Editor allows the creation from blank or from a previous file of a new user interface. Components can be added and the Alignment Tool allows movements in both directions, defines distances and orientation. It is also possible to program new functions for each GUI component. The programming is made in the M-file Editor.

3.0 The Graphical Application

The partial differential equation that describes the main mechanisms of heat transfer in a fluid [5] was chosen (Equation 1), because this CFD field is the main subject of various post graduation studies at the School of Engineering, University of Minho.

$$\rho C_p \frac{\partial T}{\partial t} = k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) + u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} \quad (1)$$

This PDE equation (1) has 4 terms, which describe the energy accumulation along the time in the domain studied, the energy transfer by conduction in both space directions and the last two terms describe the transfer by convection, in each direction, respectively.

3.1 Algorithm

In solving Equation (1), the following steps are taken:

1. *Definition of the physical and numerical parameters.* The physical parameters include the domain geometry, the physical proprieties of the working fluid. The problem can be transient (it means that changes in time are studied) or steady state (the solution has no changes with time and so a temperature field is calculated).

2. *Grid generation.* Because the numerical solution always gives a discrete solution, the user must define the number of points (x, y) where the solution must be found.

3. *Equation discretization.* The time derivatives at Equation (1) are approximated by algebraic equations and different integration schemes can be applied [6]. The discretization of

the domain uses different methodologies to solve the PDE. The Finite Volume method is used [7].

4. *Boundary conditions.* The physical domain to be solved is dependent of the physical transfer phenomena involved but also its final field temperature is dependent upon the boundary conditions. The user must introduce these values.

5. *Solution of the algebraic equations.* Once the discretized equation is applied to all grid points into the domain and the boundary conditions are defined, an algebraic system of linear equations has to be solved. The resulting system is solved by appropriated numerical methods.

6. *Main Results.* In the transient case, the user defines the temperature field at the start of the simulation and step by step, the numerical algorithm evaluates the temperature field. In the steady case, the user is asked to give a first approximation of the final temperature values for all points of the domain. In an iterative way, the numerical methods will converge to the solution.

3.2 Menu Definition

The options given to the user was one of the main steps to be defined: how the input was introduced and which choices were allowed. This is always a difficult step because it will define all the subsequent work and implementation. Different menus can be choosen.

3.3 Input Data

Although all fields have predefined (default) values, and so a default case can be always solved and visualized, the user can change these values. All fields are validated and stored into a data structure 'handles' which allows the user to define a particular parameter as global or local. Subsequent data introduction, the domain, the grid points and the main choices are visualized in order to be validated by the user (Figure 2). Table 1 summarizes all input values.

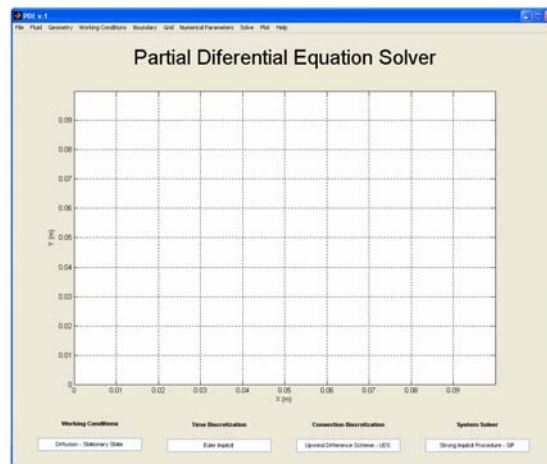


Figure 2. Input data visualization.

Table 1 - Input data

Menu	Nome
<i>Fluid Properties</i>	<i>Density</i>
	<i>Thermal Conductivity</i>
	<i>Heat Capacity</i>
<i>X Dimensions</i>	<i>X Minimum</i>
	<i>X Maximum</i>
<i>Y Dimensions</i>	<i>Y Minimum</i>
	<i>Y Maximum</i>
<i>Working Conditions</i>	<i>Time Step</i>
	<i>Maximum Number of Time Steps</i>
	<i>Number of Time Between Prints</i>
	<i>Velocity in XX'</i>
	<i>Velocity in YY'</i>
<i>Boundary Conditions</i>	<i>Temperature</i>
<i>Numerical Parameters</i>	<i>Alfa Parameter for SIP Solver</i>
	<i>Maximum Number of Iterations</i>
	<i>Error – Maximum Residue</i>
<i>Solve Problem</i>	<i>Initial Conditions</i>

3.4 Solving Step

To execute this step (Figure 3), the user is asked to introduce the initial temperature field (or for the steady state case, the best approximation to the final solution).

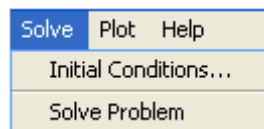


Figure 3. Solve step menu.

All the code necessary to perform the numerical calculation was introduced in MATLAB, and it is based into some Fortran routines, taken from the literature [7].

3.5 Print Results

The user can choose four different plots of the results (2D or 3D form). A plot is generated and can be exported to other applications, which improves the application. One example of a 3D plot is illustrated in Figure 4.

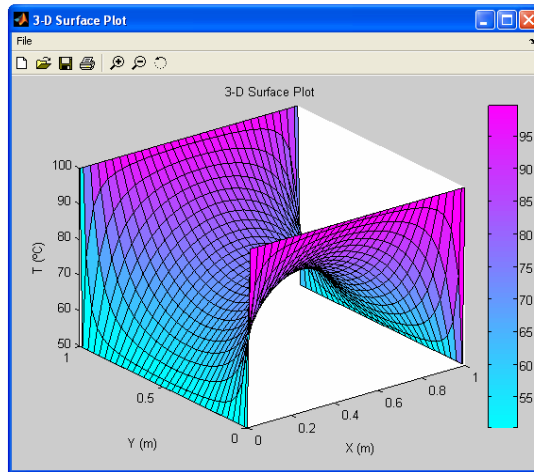


Figure 4. Example of a 3D Surface Plot.

3.6 Help Menu

All steps of the graphical application have been explained in a help file (Contents) in HTML.

4.0 Results and Discussion

Different tests have been made to the graphical interface. The influence of the different numerical parameters was studied, the various user defined options were tested and the numerical code was validated with standard cases.

Here, some results are presented. They represent some of the cases studied during the lectures with the post graduation studies.

The first corresponds to a 2D case, in a square domain (1 m x 1 m). The material used was steel, the boundaries of the domain were kept at constant temperatures (20° C and 50° C), and only diffusion transfer were studied.

In this case, the user could introduce a first approximation to the final steady state field temperature and the results are plotted in Figure 5.

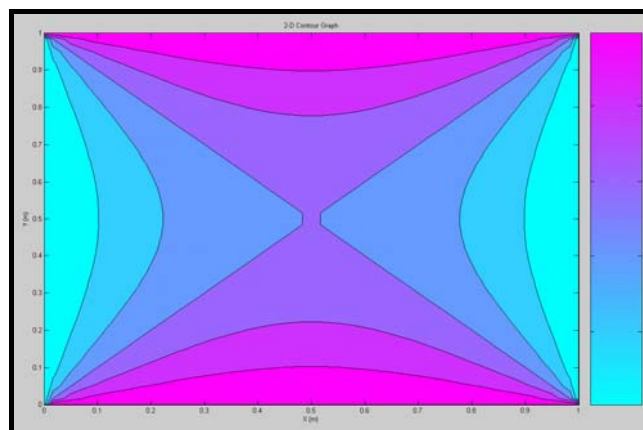


Figure 5. Results for a steady state case.

It is important for teaching proposals, to compare results obtained with different numerical methods (discretization method and algebraic system solution), as well as, different numerical parameters, in the same software (MATLAB).

Figure 6 presents the output obtained with the PDE Toolbox. The finite element method used in this software does not look so accurate in this standard symmetrical case.

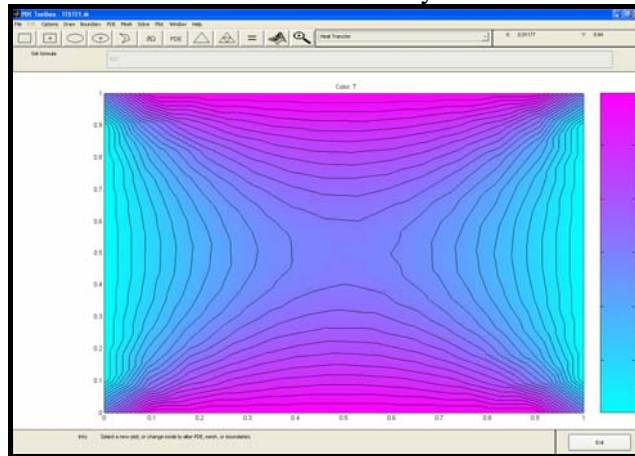


Figure 6. Results for a steady case with MATLAB.

In the transient case, the user can study the system evolution along the time and can choose the time step for the numerical integration and the interval for the printouts.

Figure 7 presents three temperature fields at three different times.

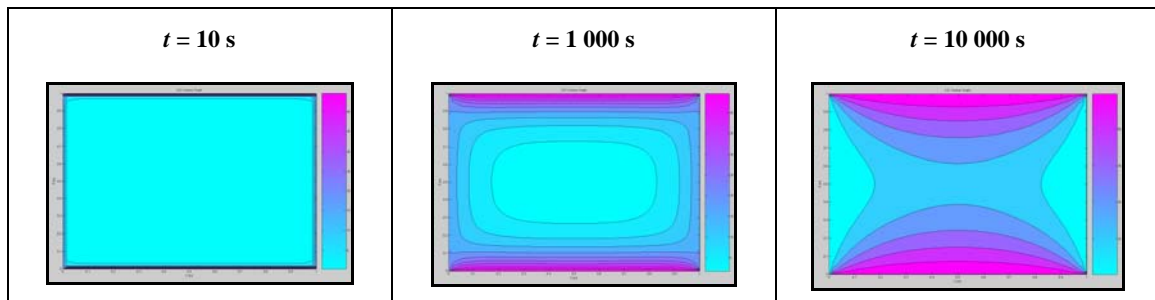


Figure 7. Results for transient case.

For a fluid (water in this case), some results are now presented when the convection mechanism is allowed (Figure 8). All the boundaries are now at different temperatures and the fluid has a velocity component ($u=1.e-6$ m/s).

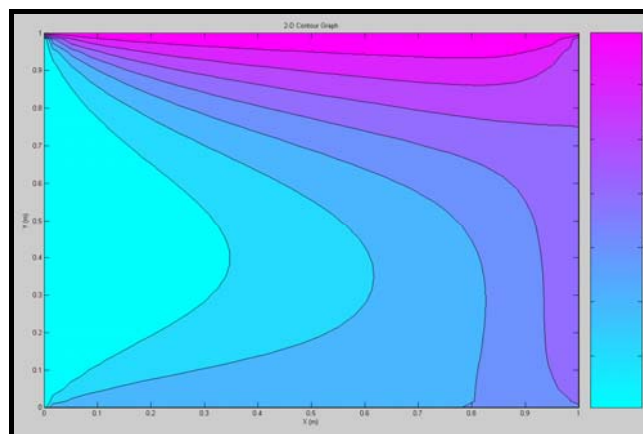


Figure 8. Results with convection effects.

5.0 Main Conclusions

The graphical application PDE v.1 is a friendly and interactive way of solving heat transfer problems in engineering fields.

The user can easily introduce, modify and test different problem parameters, some numerical choices and include even different terms of the same PDE equation.

Its graphical facilities allows a quick visualization of the results, exportation for other applications and the user can also save the numerical algebraic results for further calculations.

The work up to the present has already demonstrated a good versatility in solving the proposed problems.

Further implementations are taking place to include new features, but it can be conclude that it represents a good investment in research work as well as a teaching tool.

6.0 References

- [1] Maria Isabel Barros. “Modelação do Comportamento Térmico de Moldes de Injecção”. Ph.D. Thesis, University of Minho, Portugal, 2004.
- [2] Teresa Campos. “Modelação do Escoamento de Termoplásticos Fundidos em Moldação por Injecção”. Ph.D. Thesis, University of Minho, Portugal, 2004.
- [3] Tiago Pinho. “Solução Numérica de um Problema Térmico de Materiais com Mudança de Fase”. Technical Report, University of Minho, Portugal, 2004.
- [4] Marchand. “Graphics and GUIs with MATLAB”. 2nd ed. Boca Raton, CRC Press, 1999.
- [5] Senhorinha Teixeira and José Teixeira. “Métodos Numéricos em Transferência de Calor”. Technical Report, University of Minho, Portugal, 2003.
- [6] Chapra and Canale. “Numerical Methods for Engineers”. 2nd ed. New York: McGraw-Hill, 1990.
- [7] Ferziger and Peric. “Computational Methods for Fluid Dynamics”. Springer, 1999