

# Using MATLAB Component Object Model with Visual Basic Graphical User Interface (GUI): Application To: One Dimensional Diffusion Heat Transfer Equations of Extended Surface (FINS)

Mohammed Khalafalla Mohammed<sup>1</sup>, Mahir Abdelwahid Ibrahim Ismail<sup>2</sup>

<sup>1</sup>Electronic Engineering Department, Tianjin University of Technology and Education, Tianjin 300222, China

<sup>2</sup>Mechanical Engineering Department, Tianjin University of Technology and Education, Tianjin 300222, China

**Abstract:** *The main problem objects implementation of all learning projects or labs in the world is the high cost. So all manufacturer companies in the world resort to find other teaching methods which have low cost and more reliability to replace these labs. One of these methods are called simulation software which make user performs his experiments as when he uses actual labs. In this paper simple way for teaching one dimensional diffusion heat transfer equations of extended surface (fins) was made for readers or users. This way was done with Microsoft COM technology as communication instrument between Visual Basic and MATLAB application. By COM technology we can exchange our data between two applications. The application which sends the data by COM technology is called client and the other which receives data by COM technology is called server. Here Visual Basic was used as COM client and MATLAB as COM server. By this way the user does not need to know any things about Visual Basic or MATLAB just he uses them to study one dimensional diffusion heat transfer equations of extended surface (fins) when he uses Visual Basic GUI and enters his input parameters then will see the results very easy and clear. Also by this way we can offer the time need to study this case study.*

**Keywords:** Component Object Model, User Interface, Visual Basic, Heat transfer.

## 1. Introduction

COM (Component Object Model) is a Microsoft framework designed for Windows platforms for developing and integrating software components. Software components and reusability techniques have interesting advantages, as component base software engineering has shown through the last years. The most powerful idea around component-based on software, is that components can be implemented by a programmer and reused by others without having knowledge of the source code. Components are binary packages that can be deployed and further integrated with others written on different programming languages. As component selection and integration is usually an easy and well-known process, components are also called COTS (Commercial Off-The-Shelf).

Visual Basic software is very famous for using Graphical User Interface (GUI). Because of its properties that seems very easy and clear. An important characteristic of Visual Basic is related to the multitasking operating environment of Windows. Here Visual Basic application is used to make MATLAB application running at the same time. In this case MATLAB is running in special case called MATLAB Automation Server. MATLAB is a technical computing environment for high performance numeric computation and data visualization and its use has become popular in analytical purposes like MATLAB integrates numerical analysis, matrix computation, signal processing and graphics in an easy-to-use environment that allows the user to do many matrix computations employing language commands that look very much like standard linear algebraic notation. In our program developed Visual Basic routines control the data acquisition and uses COM server to send data to

MATLAB for calculations and graphical visualization. Finally we receive results from MATLAB in Visual Basic Graphical User Interface (GUI). At this time is possible to improve the Visual Basic programs using other Windows applications.

It is very important when we do not have enough time to learn using software. So using (GUI) makes software very easy and decrease the time that need to learn it. This fact implies that there will be less time devoted to the software applications being covered. In MATLAB we can make GUI non-visually by writing MATLAB code which determines the properties of the controls that it seems very difficult and complicated. In Visual Basic we can make GUI visually by choose the control we need from the control menu and double click with left mouse on it then adjust its properties in the properties window that it seems very easy and simplify.

Our GUI is constructed from only one form which contained six labels control to define the names of textboxes control, four buttons control to controlling the execution of the program, two picture boxes which contain the case study table and figure of heat transfer through the fin. Also it contains four textboxes control which will use by the user to put his values of input variables.

## 2. Microsoft COM technology

A COM component is an instance of the component object class that runs on the COM server and is accessible from a variety of clients. There are several platforms that can serve COM objects and many clients that can use them. MATLAB COM components are very useful to integrate tasks

implemented on this language and exported to others applications. These components can be used later in Microsoft Office Applications or even other MATLAB clients, in local or distributed applications.

### 3. MATLAB COM Integration

The Component Object Model, or COM, provides a framework for integrating reusable, binary software components into an application. Because components are implemented with compiled code, the source code may be written in any of the many programming languages that support COM. Upgrades to applications are simplified, as components can be simply swapped without the need to recompile the entire application. In addition, a component's location is transparent to the application, so components may be relocated to a separate process or even a remote system without having to modify the application. Using COM, developers and end users can select application-specific components produced by different vendors and integrate them into a complete application solution.

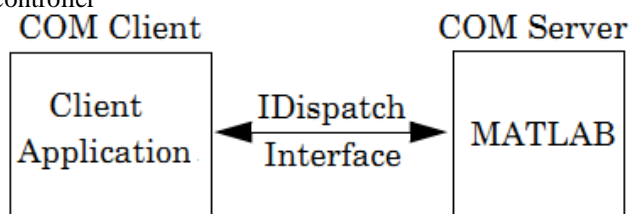
### 4. MATLAB programmatic identifiers

MATLAB provides three programmatic identifiers to access COM servers:-

- **MATLAB.Application:** This identifier is used to start a MATLAB Automation server on an independent window. A command window will appear to enter commands.
- **MATLAB.Autoserver:** A MATLAB Automation server is started on an independent command window.
- **MATLAB.Desktop.Application:** With this configuration, a MATLAB full desktop is started. By running the registry tool in Windows, we will be able to find out the identifiers for accessing the COM servers.
- **-MATLAB.AutoServer.Single:** The single attribute will make the server to run on an exclusive mode. Otherwise the server will starts on a shared configuration. In this paper we used first one (MATLAB.Application).

### 5. Client Application and MATLAB Automation Server

MATLAB operates as the Automation server in this paper and it controlled by Visual Basic to be an Automation controller



**Figure 1:** MATLAB COM Server and COM Client

MATLAB Automation server capabilities include the ability to execute commands in the MATLAB workspace, and to get and put matrices directly from and into the workspace. We can start a MATLAB server to run in either a shared or dedicated mode. Also we have the option of running it on a local or remote system. To create the MATLAB server from an external application (visual basic) program, we use the

appropriate function from visual basic to instantiate the server. Here we use the Create Object function in Visual Basic. For the programmatic identifier, specify MATLAB. Application to run MATLAB as a dedicated server.

### 6. MATLAB Automation Server Tasks

MATLAB Automation provides a number of tasks that we can do them in MATLAB or Visual Basic to manipulate data in the MATLAB server. These are:-

- Executing Commands in the MATLAB Server
- Exchanging Data with the Server
- Controlling the Server Window Terminating the Server Process

### 7. Client-Specific Information

In COM servers we find available interface methods. The most useful are:

- **MLApp.Execute().** This method executes a MATLAB command on the COM server.
- **MLApp.Feval().** This method evaluates a MATLAB function.
- **MLApp.GetFullMatrix().** This method is useful to copy a matrix from MATLAB workspace.
- **MLApp.PutFullMatrix().** This method is useful to copy a matrix into MATLAB workspace.
- **MLApp.PutWorkspaceData().** This method is used to put variables into MATLAB workspace.
- **MLApp.GetWorkspaceData().** This method is used to read variables from MATLAB workspace.
- **MLApp.Quit().** It must be used to force exit when the host application finalizes.

### 8. MATLAB ActiveX Automation Method

This section lists the method that is supported by the MATLAB Automation Server. The data types for the arguments and return values are expressed as ActiveX Automation data types, which are language independent types defined by the ActiveX Automation protocol. Like BSTR is a wide-character string type defined as an Automation type, and is the same data format used by Visual Basic to store strings. Any ActiveX-compliant controller should support these data types, although the details of how we declare and manipulate these are controller specific.

### 9. BSTR Execute in BSTR Command

This command accepts a single string (Command), which contains any command that can be typed at the MATLAB command window prompt. MATLAB will execute the command and return the results as a string. Any figure windows generated by the command are displayed on the screen as if the command were executed directly from the command window or an M-file.

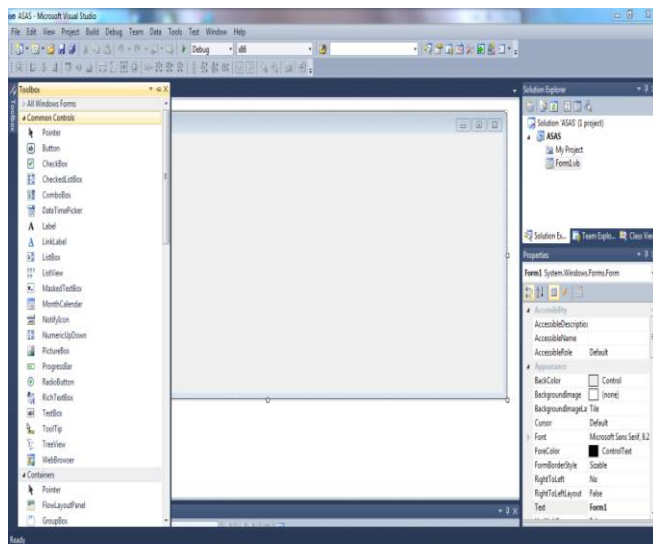
### 10. Client Specific Information

For Visual Basic.net Clients Data types for the arguments and return values of the server functions are expressed as

Automation data types which are language independent types defined by the Automation protocol. Like, BSTR is a wide-character string type defined as an Automation type and is the same data format used by Visual Basic to store strings. Any COM compliant controller should support these data types, although the details of how we declare and manipulate these are controller specific.

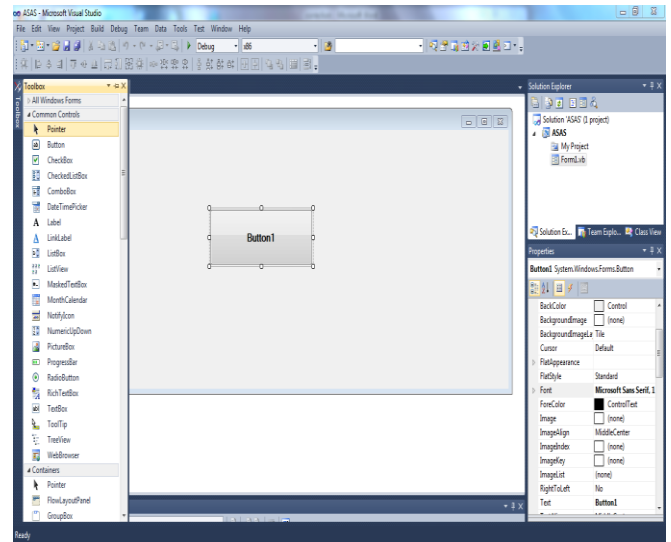
## 11. Visual Basic Graphical User Interface (GUI)

Visual Basic was designed for making GUI's. Figure (1) is a screen shot of a new visual basic project. The object in the center is called a form. On the left is a tool box and on the right is a box which contains all the properties of the form. In the toolbox are items (also called controls) that can be placed on the form. These controls contain a command button, a text box, a label box and many other things. Placing any one of these items on the form is done by make double clicking on the item in the toolbox or click on it then dragging it to the correct position of it.



**Figure 2:** Visual Basic.net start-up screen

In figure (2) we can see a control button has been placed on the form. For setting the properties of it we can go to the properties window on the right of the form which contains the form properties and any object placed on the form. The flow of execution is dependent on the action the user takes. When the user presses a button on the GUI, the program handles that event.



**Figure 3:** Button Box

Each object control has an event (may be more than one) that can be done with it. In order to write event handler with any object just make double clicks on it to bring code widow for it. Also we can add a separate code module to the form to holds subroutines and declaration of the variables.

In figure (3) we see our GUI which contains label box1 on the left side that show the procedures of using GUI, picture box1 under the label box1 include the table of case study of our application, picture box2 behind the label box1 shows small figure of Parameters of heat transfer through the fin to help user to understand something about variables. Label box2 show the address of the GUI topic. On the right side of GUI we find label box3, label box4, label box5, label box6 show the name of application parameters that must be choose by users. Each label of them has a text box in front of it where user can put his own values in them. Also GUI include four command buttons, one used for sending data to mat lab for executing then get the result back to the visual basic GUI. The second one used to clear the old data and re-enter new ones .The third one for clear all data when the user want to change to new study case. The last one for ends the task or end the visual basic running.



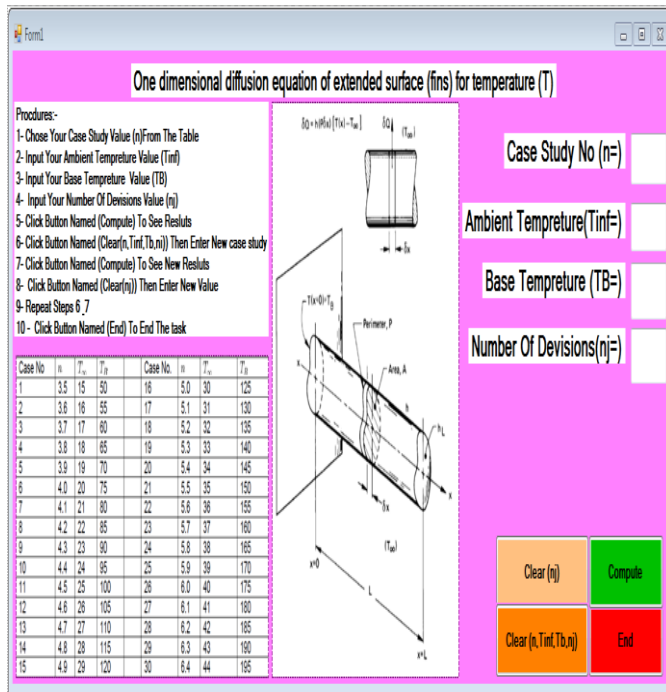


Figure 4: Graphical User Interface

## 12. Procedures and Results

- Open –start menu-choose Microsoft Visual Studio 2010- then click left mouse on it (do not run MATLAB application).The visual basic window will appear as below:-

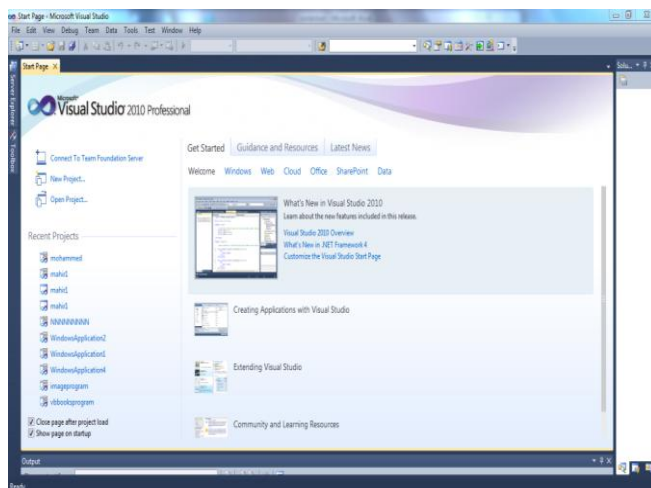


Figure 5: Visual Basic.net project start window

- Run the program (by click left mouse on run icon). The program (GUI) window will appear as below:-

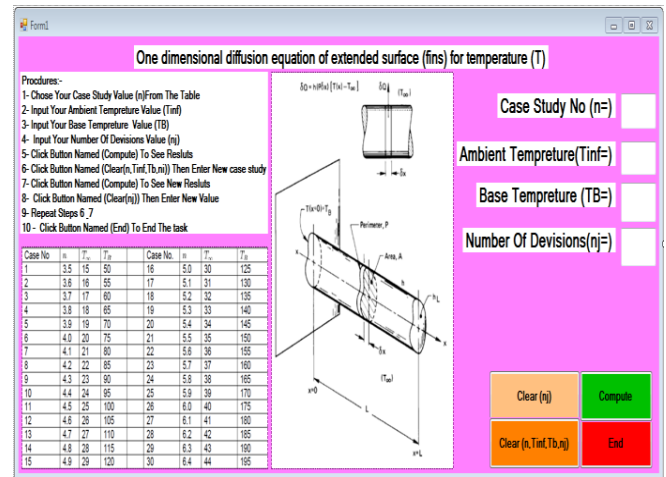


Figure 6: Visual Basic.net project GUI window

- Click the icon to run the GUI program then enter the four (4) values of study case variables (n, Tinf, Tb, nj) in the coordinating text boxes .Here the values were chosen to be (n = 4, Tinf = 20, Tb = 75,nj = 10). The program (GUI) running window will appear as below:-

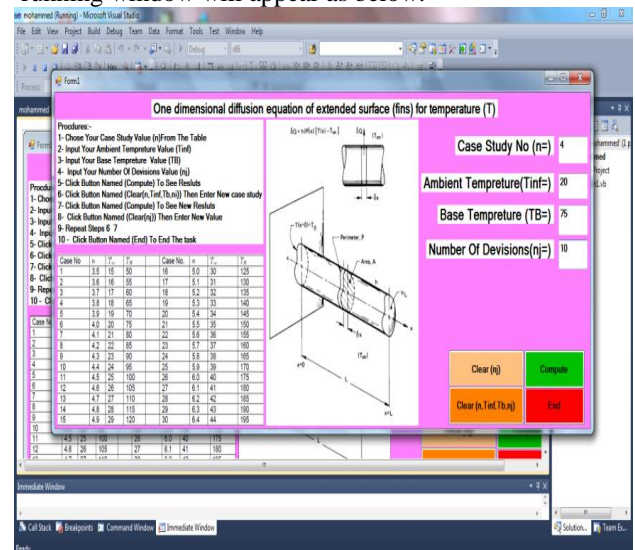
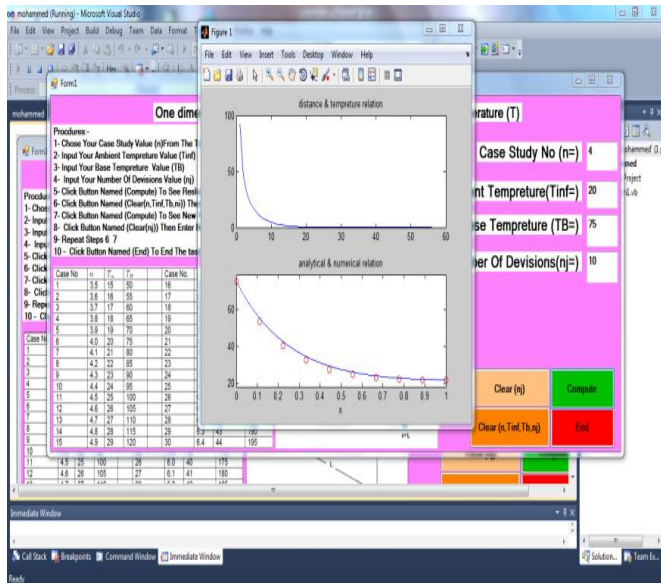


Figure 7: Visual Basic.net project running GUI inputs window

- Now this data of one dimensional diffusion equation of extended surface (fins) must be send from visual basis application to MATLAB application for execution then get the results back from MATLAB application to visual basic application. For this purpose special command was make in this GUI which named (Compute) and that took green color. This command contains COM code to perform this function. So click left mouse on command button named (compute) to see the results according to inputs (the figure will appear immediately) then look to the results. The results window will appear as below:-



**Figure 8:** Visual Basic.net project running GUI results window

For more development of using this GUI another two command buttons were added to perform additional option functions. One command button was named (clear (nj)) and it used to clear old (nj) then replaced it by new one to see another results. This function was done by clicked left mouse on it. Next command button was named (clear (n, Tinf, Tb, nj)) and it used to clear old values of (n, Tinf, Tb, nj)) then replaced them by new ones to see another results. This function can be done by click left mouse on it. By this way many cases can be chosen from the case study table by the users to study them. Finally to end using of the application click left mouse on command button named (end) must be done.

### 13. Conclusion

COM technology is a useful tool for integrating software from different vendors. This is specifically interesting for engineering applications, which have to integrate a great variety of software functionality, from specialized and low-level task, to intuitive GUIs. Fortunately, software component industry has provided solutions to properly combine software components to get a complete solution without implementing the software from scratch. This can be made using reusability techniques following the standards defined by the Component Base Software Engineering (CBSE).

Engineers working with MATLAB and other software can take advantage of CBSE using COM and .Net technologies from Microsoft. Furthermore, integrating different objects from different applications accelerates software development and reduces costs. So we think this paper can be considered as small simple way make the reader use it to understand how COM technology is very important for engineering application.

After we wrote this paper we expect that the interested reader can take into account component technology on their future projects, and gain effectiveness on the overall software development process.

### References

- [1] Mahir Abdelwahid Ibrahim Ismail, Mohammed Khalafalla Mohammed- Gantry Robot Kinematic Analysis User Interface Based on Visual Basic and MATLAB- International Journal of Science and Research (IJSR), Paper 482 ID: SUB151159, Volume 4 Issue 2, February 2015.
- [2] Gunderloy, M. (2001). Calling COM Components from .NET clients, In: MSDNLibrary, 23.03.2012, Available from: <http://msdn2.microsoft.com/en-us/library/ms973800.aspx>.
- [3] Huang, C. (2006). Tcom, In: Access and implement COM objects with Tcl, 23.03.2012, Available from: <http://www.vex.net/~cthuang/tcom/>
- [4] Raboso, M.; Izquierdo, A. & Villacorta J.J. (2003). Beamforming Systems Modeling using Component Reusability with XML Language, Proceedings of the International Signal Processing Conference, ISPC 2003, Dallas, Texas, USA, March 31-April 3, 2003
- [5] Raboso, M. (2007). Beamforming Systems Modeling Using XML Language, based on Software Component Reuse, ProQuest Information and Learning, ISBN 978-0-549-26134-6, USA.
- [6] Raboso M. ; Izquierdo A. ; Villacorta J. ; del Val L. & Jiménez, M. (2009). Integración de componentes COM de MATLAB/SIMULINK en el entorno CASE XBDK para el modelado de sistemas de conformación de haz. Ingeniare, Revista chilena de ingeniería, Vol.17, No.1, (January 2009), pp. 122-135, ISSN 0718-3305
- [7] The Mathworks (2012). MATLAB COM Automation Server Support, In: MATLAB R2012a Documentation, 23.03.2012, Available from: [http://www.mathworks.es/help/techdoc/MATLAB\\_external/brd0v3w.html](http://www.mathworks.es/help/techdoc/MATLAB_external/brd0v3w.html)