IMPLEMENTATION OF FULL WEB-BASED GRAPHIC USER INTERFACE PROCESSOR FOR CFD SOFTWARE

Makhsuda Juraeva*1, Evgeny G. Ivanov*1, Dong Joo Song*2

웹 기반 CFD s/w용 GUI 프로세서의 구현

막슈다 쥬래바, 에브게니 이바너프, 송동주

The preprocessor - solver - postprocessor software for 2D/Axisymmetric CSCM Upwind Flux Difference Splitting Navier-Stokes code has been developed for undergraduate educational purpose. This computational fluid dynamics (CFD) software allows students to setup, solve, visualize and control dynamically server for their own fluid problems via Internet. The preprocessor is capable of generating geometry and grid, initial solution data and required solver control parameters. The postprocessor shows vector plot and contour plot with different options while residual plot shows root-mean-square (RMS) error history graphically and retrieves the data from solver interactively. Special feature of the preprocessor is grid generation part which is based on MFC/Visual C++ application and FORTRAN single block grid generator process. Many users can access solver via Internet from client computers and solve desired problems using locally installed pre- and postprocessor and remote powerful solver part.

Key Words: Preprocessor; postprocessor; web; GUI.

1. Introduction

Graphic User Interface (GUI) has significant importance for pre- and postprocessor. And common users can use pre- and postprocessor and Navier-Stokes solver for their own problems. The preprocessor is a part of software, which provides convenient working environment, where user can generate geometry, computational mesh and required input data. The Postprocessor is bounded in a program along with preprocessor and solver. It visualizes the data produced by the solver.

Non Internet-based preprocessor [1] and postprocessor [2] for 2D/Axisymmetric CSCM Upwind Flux Difference Splitting Navier-Stokes [3]code were developed as a first step. So that users could solve problems on their local computers.

Main goal of the programs development was creating software accessible by the largest possible number of users to solve their CFD problems. Internet was selected as development environment.

As a basis, preprocessor and postprocessor are going to be used on the client computer and the solver part is supposed to be on the server computer.

To implement the strategy mentioned above, it is necessary to extend full process ability by adding network functions for data exchange between client and server part.

The main advantages of the developing Internet-based CFD software are:

- Easy access by the large number of users with relatively slow computers to server;
- Convenient user interface, which minimizes both the learning time and specialized knowledge required;
- Access for academic users (students and researchers) which means that it can serve as a teaching tool for CFD.

*1 영남대 대학원 기계공학과
*2 영남대 기계공학부
E-mail: *1: mjuraeva@yumail.ac.kr, ievg@yumail.ac.kr
*2: djsong@yu.ac.kr

2. Strategy of implementation

There are a lot of preprocessors and postprocessors in many different famous commercial packages which can be good examples of preprocessor and postprocessor, but several of them were mainly considered as references in this study; Gambit [4], CFD-FASTRAN preprocessor [5] and Tecplot of AMTEC [6]. The preprocessor - solver - postprocessor software can be a powerful educational tool, giving students an appreciation for how fluid behaves under different conditions.

2.1 Visual C++ MFC and GUI environment

One of the main features of the developing full processor is a convenient Graphic User Interface. Visual C++ and MFC provide programming tools and automatic techniques for rapid visual application development. An application created with Visual C++ and MFC will automatically generate most of its own windows, handle its own messages and do its own drawing [7].

2.2 FORTRAN 77 and Visual C++ 6.0 collaborative work

As was mentioned above, some of preprocessor steps were usually done manually or with help of additional programs. The most important part of it was FORTRAN single block grid generator. It has been used in the new developed Windows application.

By calling spawn function [8] it is possible to execute a new process. In this case FORTRAN single block grid generator should be executed.

2.3 Multi block grid generator

Usually, single block grid is not adequate for solving realistic problems. Existence of multi block option for grid generation crucially changes number of available types of grid which can be generated by preprocessor. FORTRAN grid generator is a single block grid generator. It means, that user has to generate several single blocks and join them together manually to solve complex flow problems. It is another objective of the work to make this process work automatically.

2.4 Contour plot

Make two developed function for look starting point of required points on the boundaries of object and for find next required points. Mesh lines search method is going to use to find required points. Required point coordinates are found along the grid lines by using first order linear interpolation [9]. Following condition is used to find required point over the problem object:

$$(Pt\ (i,j)point)\cdot (Pt\ (i+1,j)point) = 0$$

where: $Pt\ (i,j)$ and $Pt\ (i+1,j)$ are the flow variable values at i and j mesh indexes; point is required point which is used to draw contour line. Required point is chosen automatically by adding the contour line period to the minimal contour value of flow character. Contour line period is computed by adding maximal contour line value to minimal contour line value and divided by amount of the given contour lines.

2.5 Residual plot

At the end of solver iterations, results of each checked variables are computed and stored, thus the convergence history is recorded. This history is saved at the data file. While solver carrying out computations, Residual plot reads this data file. Then program shows data graphically and closes data file. After this, application opens data file again and draws the error history with new data until computation has been completed.

2.6 Data exchange via Internet

CFD problems are traditionally slow to converge, and require a significant amount of computer memory. They are normally run on powerful computers. Preprocessor and postprocessor should operate on the user computer, but all computational work must be done by computers managed by server.

Preprocessor provides data for solver. This data can be classified into three groups: grid with geometry, initial data, and control parameters data. It is necessary to provide all information for successful solver operation.

All data are provided in the form of files: fort.1 includes mesh data and initial solution data, fort.5 control parameter data.

User provides preprocessor data, computers, managed by server, perform calculations and preprocessor receives solution back on the local personal computer.

To extend developed preprocessor and postprocessor for network data exchange WinInet and WinSock MFC classes

are going to be used first. Thus, preprocessor can send input solver data to server and postprocessor can receive residual information and solution data.

3. Using present software A sample problem

To make the software a powerful educational tool for engineering students, there is an instruction. Instruction teaches students how to use the software through the setup and solution of some CFD problems.

The Navier-Stokes simulation of supersonic flow over 16 degree compression ramp has been performed [10]. A 16 degree compression ramp problem was examined with inflow conditions.

For each problem users are required to go through the following steps:

1) Grid generation option
2) Multi-block option
3) Flow definitions
4) Initial and Boundary conditions
5) Solver controls and connection to Solver
6) Analyzing results (postprocessor)

Grid generation option. The grid generation part is the most important step in preprocessor work. If grid generation part has many options and convenient interface, user can create desired mesh where he is going to obtain solution.

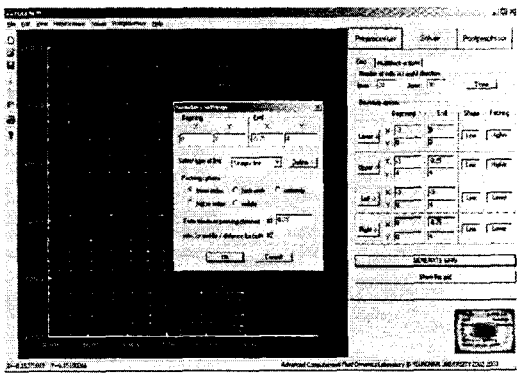

Fig.1 Grid generation part in the application

User has to follow certain scheme to create grid. First of all, user is asked to define number of cells for both directions then choose boundary and set for it: coordinates of beginning and end, shape of the line, type of packing. After software has all information about each boundary user can generate grid. Figure 1 shows how it is implemented in application.

Multi-block option. Multi-block option should be enabled for multi-block grid generation. It means that after generating several blocks they are joined together into one whole grid. For that purpose in Boundary Conditions part "interface" type of boundary is added.

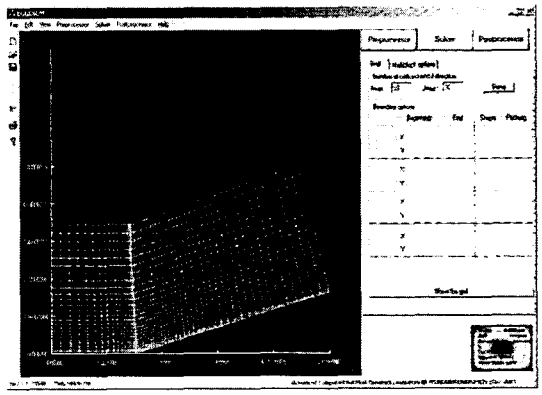

Fig. 2 Compression ramp: summation of two blocks

It indicates boundary interfacing with another block. Then, using multi-block option it will be connected into one grid as shown in Fig. 2. Next steps are Flow definition part, Initial conditions and Solver controls options.

Flow definition option. After connection has been completed user can start to define his problem. Flow definition part in the developed preprocessor is for flow properties settings.

Boundary conditions option. Figure 3 shows list of available options for Boundary Conditions part and how it is realized in the application.

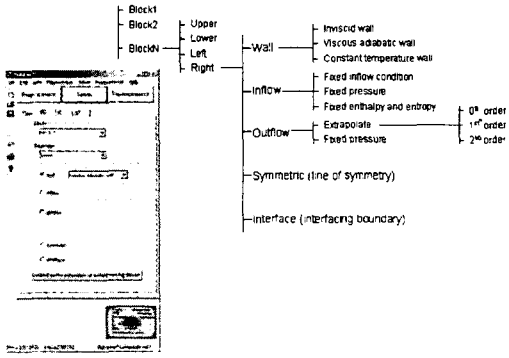

Fig. 3 Boundary Conditions part in the application

Initial conditions option. Initial conditions can be generated in the following ways: it can be taken from file, generated by using reference values or it can be entered manually by user. In the lower part of the panel there are edit boxes which display density, U-velocity, V-velocity and pressure
Solver control option. Solver control part includes parameters to control solver work such as iterations number, order of accuracy, initial and final CFL number.
Network Solver Option. Data exchange over Internet was carried out by using two main principles: File Transfer Protocol (FTP) client and server and so called "messages"passing. FTP is used for file sending and reading and "messages" helps to both server and client parts to determine when and what action must be taken. Figure 4 shows how clients and server interact with each other.
In case of network solver, computation will be performed on remote server computer.

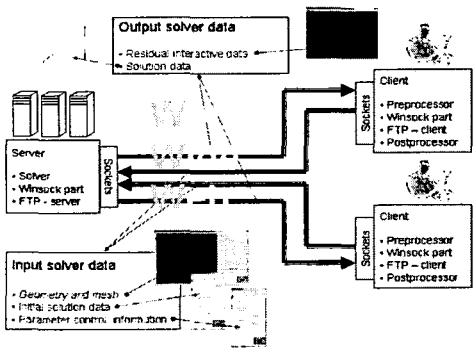

Fig. 4. Scheme of clients-server (pre-post-processor - solver) interaction

User has to connect to server and control all options dynamically. When user connected, solver starts listening all messages from user. Data files have to be sent for execution. And user executes solver from network. There is "Show Residual" option to see residual plot. After calculations have been performed user can see results by taking data file from server.
Residual plot option. Residual plot opens data file and draws the error history until computation has been completed. There is time for all iterations until computation results are obtained. During the plotting, vertical axis shows error value while horizontal axis shows iterations (Fig. 5). Axis values are renewed with each iteration. Residual plot in the program draws lines in the plot window with the common minimum and maximum error values. The program does not have capability of residual monitoring yet.

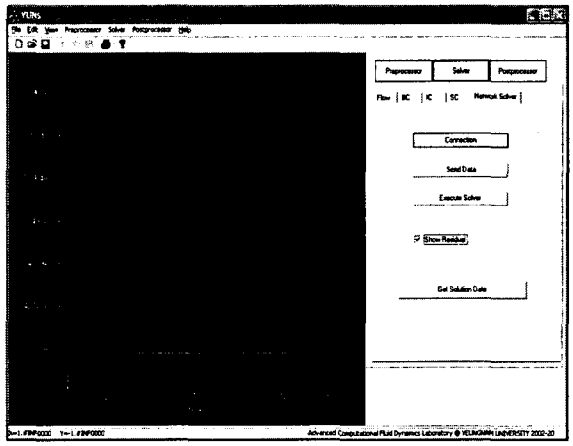

Fig. 5 Residual Plot

When calculation is over user can get solution data back from server computer and postprocessor part will show solution in different forms and views.
When user opens required data file from /File/Open submenu, lines are drawn in the plot window and mesh plot appears in the plot window for contour plot and vector plot.
User can manage other options from Side bar or menu. The present software has the same data format as Tecplot program which is useful to compare results.
XY plot option. XY plot option can be used to plot result lines. Plot lines differ from each other with colors. Program initially sets the ranges on the X and Y axes, so that user can see all data graphically. Developed function finds data limits and chooses axis ranges.
Vector Plot option. Vector Plot is created by two velocity variables associated with vector components. Selection of vector length makes it possible to draw several overlapping vectors with colors corresponding to their lengths. Filled arrowhead vectors for Velocity Vector Plot are drawn. User modifies vector length from Vector Scale option. Figure 6 shows postprocessor with different options.
Contour plot option. A set of points is irregularly distributed over an object region. Points required through the object region are found and connected into lines. Mesh line search method includes following method if there is required point on the one of a mesh side and next required

point must be one of the other three sides of mesh [11]. If first developed function can not find starting point of required points on the object boundaries it continues looking for mesh next to boundaries. These two developed functions also can find required points of circle contour lines. The program selects amount of the given contour lines automatically and while using the program, user can modify this amount manually. Contour Plot lines are identified by colors. Colors of these lines are chosen by value of flow character and their tints are distributed between three colors: red, green, and blue.

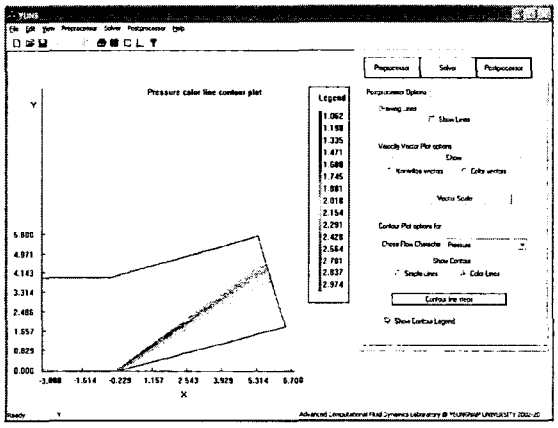

Fig. 6 Pressure contour plot for sample problem

4. Conclusions

Present Internet-based software for 2D/Axisymmetric CSCM Upwind Flux Difference Splitting Navier-Stokes code has been developed in ACFD laboratory for undergraduate CFD educational purpose. The software provides students to use their Fluid Mechanics knowledge to solve a flow problem and visualize the solution. After a short introductory session, undergraduate engineering students are able to solve relevant and interesting Fluid Mechanics problems.

To improve productivity and make easy access by a group of users, network access and powerful solver computer (server) were used. Users can access server (solver) via Internet from client computers and solve desired problems using locally installed pre- and postprocessor user interface. Present software is still in an early developmental stage and requires additional work before it reaches its full potential as a teaching tool.

5. Further works

The following issues could be further worked:

- Creating of OS independent full web based GUI processor with C#
- Improving of multi-user access and computer resources management.

References

[1] Evgeny G. Ivanov, Dong Joo Song, Implementation of Preprocessor for CSCM code by using Graphic User Interface. *Korean Society of Computational Fluids Engineering,* 2003, pp. 69-75.

[2] Makhsuda Juraeva, Dong Joo Song, Implementation of Postprocessor for CSCM code by using Graphic User Interface. *Korean Society of Computational Fluids Engineering,* 2003, pp. 76-81.

[3] C. K Lombard, J. Bardina, E. Venkatapathy, and J. Oliger, "Multi-Dimensional Formulation of CSCM An Upwind Flux Difference Eigenvector Split Method for the Compressible Navier-Stokes equations", 1983, AIAA-83-1895.

[4] FLUENT, 1998, Fluent Inc.

[5] CFD-FASTRAN. User's Manual. Version 2002, CFD Research Corporation.

[6] Tecplot. Interactive Data Visualization for Scientists & Engineers. User's manual, Ventura Software, Inc.

[7] John E. Swanke, *Visual C++ MFC Programming by Example,* 1999, CMP Books, CMP Media, Inc.

[8] MSDN Library, July 2001.

[9] Steven C. Chapra, Raymond P. Canale, *Numerical methods for engineers,* McGraw-HILL International editions.

[10] Sang Dug Kim, 1999, "Performance Analysis of Three-Dimensional Transonic Centrifugal Compressor Diffuser.", 1999. Ph.D. Thesis.

[11] L.G. Beliovskaya, N.A. Bogomolov, and A.D. Kovalev, "Construction of isoline maps of functions of two variables".