

CFD용 전처리장치 개발

김 사 량*

강릉원주대학교 기계자동차공학부

DEVELOPMENT OF THE PRE-PROCESSOR FOR CFD

S.R. Kim*

Gangneung-Wonju Nat'l Univ., School of Mechanical and Automotive Engineering

The Pre-processor program for CFD is being developed using wxWidgets and OpenGL libraries. This program can be run on both Windows and Linux operating systems. Undergraduate students and beginners can learn and use this very easily by menu and templates. Until now, structured mesh can be created in Cartesian or Cylindrical coordinates. This program will be used easily to make various type of meshes using templates.

Key Words : 전산유체역학(CFD), 전처리장치(Pre-processor), 격자(Mesh), 그래픽 유저 인터페이스(Graphic User Interface), 멀티 플랫폼(Multi-Platform)

1. 서 론

최근 들어 전산유체역학(Computational Fluid Dynamics, CFD)을 이용하여 문제를 해결하고자 하는 수요가 많이 늘고 있다. 전통적인 기계, 항공, 조선분야 뿐만 아니라, 건축, 토목, 전기, 전자, 해양, 대기 등의 분야에서도 CFD를 도입하여 실험을 대체하고 있는 실정이다. 경이적인 컴퓨터 하드웨어의 발달과 새로운 수치해석 알고리즘의 개발로 기존에는 풀기 어려웠던 문제들도 해결의 실마리를 찾고 있다.

유체공학 분야에서는 복잡한 비선형 미분방정식인 나비에-스톡스(Navier-Stokes) 방정식과 난류모델들, 그리고 여러 가지 부가적인 방정식들을 과거보다 빠른 시간 내에 풀 수 있게 됨에 따라 CFD가 점점 많은 역할을 하고 있다. CFD에서 유한체적법(Finite Volume Method, FVM)이나 유한요소법(Finite element Method, FEM) 같은 방법을 사용하는 경우에는 유동장을 수많은 격자(Mesh, Cell, Grid, etc.)로 형상을 만들어서 계산을 수행하게 된다. 그러나, 전체 해석수행시간에 비하여 격자를 만드는 작업(격자 모델링)에 오히려 더 많은 시간을 소모하는 경우도 있게 된다. 따라서, 유동장을 빠르고 정확하

게 해석하는 프로그램(Solver)의 개발도 중요하지만, 쉽고 빠르게 격자를 생성할 수 있는 프로그램, 즉 전처리장치(Pre-Processor)가 반드시 필요하다고 하겠다.

컴퓨터 하드웨어 성능의 고속화와 프로그램 성능을 최적화하여 복잡한 형상의 유체역학문제를 CFD로 푸는데 있어, 형상을 쉽고 빠르게 모델링할 수 있게 되면, 유체역학적 문제의 쉬운 해결 및 관련제품의 설계 및 생산에 들어가는 시간적인 절약에 도움이 되어, 생산원가 절감 및 국제경쟁력 확보에 큰 기여를 할 수 있다.

이미 외국에서는 이러한 프로그램들을 많이 개발하여, 전처리장치, 해석프로그램, 후처리장치를 한데 묶어서 수만달러에 달하는 비싼 가격에 판매하고 있으며, 그 수요는 계속 늘어가고 있다. 그러므로, 기술적으로는 CFD 프로그램의 효율적인 사용 및 결과의 제시(Presentation)를 위하여, 경제적으로는 외국산 소프트웨어의 수입대체 및 설계기술의 확보를 위하여, 학술적으로는 더 정밀하고 신뢰성 있는 해를 얻기 위하여 값싸고 사용하기 쉬운 3차원 전처리장치를 개발할 필요가 있다고 하겠다. 이러한 프로그램이 어느 특정한 컴퓨터 하드웨어나 소프트웨어 운영체제(Operating System, OS)에서 작동하는 것이 아니라, 우리가 사용할 수 있는 가능한 모든 컴퓨터와 OS에서 동작하게 하기 위해서 멀티 플랫폼(Multi-Platform, 주로 Windows와 LINUX에서 호환가능, Mac도 가능)을 지원하는 프로그램을 개발을 하는 것을 본 연구의 목적으로 한다.

Received: May 2, 2013, Revised: June 20, 2013,

Accepted: June 21, 2013.

* Corresponding author, E-mail: dearksr@gwnu.ac.kr

DOI <http://dx.doi.org/10.6112/kscfe.2013.18.2.072>

© KSCFE 2013

이미 우리나라에서도 서강대학교 허남건 교수 연구실에서 CFD 프로그램 개발[1-3]의 일부로 Microsoft사의 MFC기반의 전/후처리장치에 대한 개발이 시작되어 현재까지도 계속 진행되고 있으며, 국민대학교 명현국 교수 연구실에서도 vtk 라이브러리[4]를 사용하여 객체지향프로그래밍을 이용한 CFD 프로그램[5-8]을 개발하고 있다. 또한, 충남대학교 김병수 교수 연구실에서는 자바애플릿을 이용한 격자생성 프로그램[9-11]도 개발하고 있다. 그리고, 서울대학교 김찬중 교수는 M#cfid[12]라는 Microsoft사의 MFC기반의 소프트웨어 패키지를 상용을 목표로 개발하고 있다.

본 논문에서는, 지난 몇 년에 걸쳐서 저자가 개발하고 있는 전처리장치 프로그램[13,14]의 내용을 설명하도록 한다.

2. 연구방법 및 내용

2.1 연구방법

본 연구에서는 Windows나 Linux, 그리고 Mac OS/x에서도 사용이 가능하도록 하기 위해서, Microsoft사나 Apple사에 종속되지 않는 프로그램을 만드려고 노력하였다. 저자의 첫 전처리장치 프로그램은 SGI사의 Indigo2 워크스테이션에서 제작하였으며, 모든 작업에 Indigo2 머신에서 실행되는 프로그램을 사용할 수밖에 없었다. 그 이후에 마이크로소프트사의 MFC 라이브러리를 사용한 프로그램을 작성하였는데 이 프로그램은 Windows에서 밖에 실행되지 않는다. 그래서 어느 한 회사의 하드웨어나 프로그램에의 종속을 탈피하기 위해서, 일단 첫 번째로 Microsoft사의 MFC 라이브러리를 사용하지 않고 프로그램을 작성할 수 있도록 구상하였다.

Table 1에 몇 가지 GUI의 특성을 비교하여 나타내었다. 본 연구에서는 프로그램의 기본 골격을 만들 수 있는 그래픽 유저 라이브러리(GUI)로서 wxWidgets[15]를 사용하도록 선택하였다. wxWidgets GUI 라이브러리는 개발 초기에는 wxWindows라는 이름으로 불렸던 GUI 라이브러리인데 이미 십여년 이상 잘 알려져 있고, 여러 소프트웨어를 만들기 위해 사용되고 있는 공개 프로그램이다. 여러 운영체제에서 사용할 수 있기 때문에 Cross-Platform GUI Library라고도 불린다. 관심이 있는 연구자는 관련 홈페이지[16]를 참고하기 바란다. 입력장치로는 마우스와 키보드를 병행하여 사용하도록 하고,

Table 1 Comparison of GUIs

GUI	Microsoft MFC[1-3]	WxWidgets[16]	vtk[4]	Java applet[9-11]
O/S	Windows only	Cross-Platform (Windows, Linux, MacOS, Unix)		
Language	C, C++, C#	C++, python, etc.		Java
Compiler	Visual C++	Visual C++, gcc, MinGW, etc.		Javac

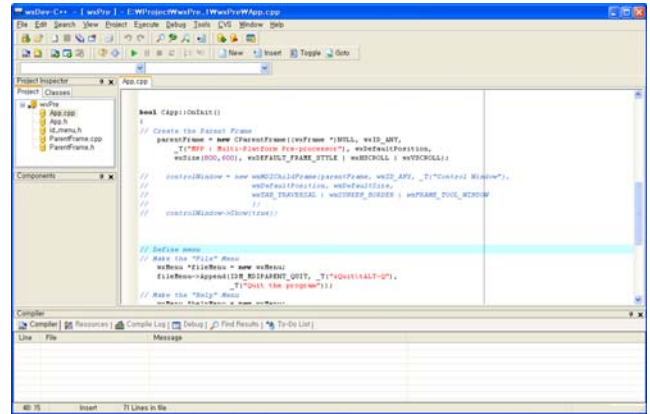


Fig. 1 An example of using wxDev-Cpp IDE

프로그램 실행시 대화형 창을 이용하여, 가능하면 키보드를 적게 사용하고 마우스를 활용하여 대부분의 기능을 사용할 수 있도록 전체 구조를 설계하였다.

3차원 그래픽 라이브러리는 SGI (Silicon Graphics Inc.)에서 개발한 OpenGL[17-20]을 사용하도록 한다. SGI사에서는 오래 전에 GL이라는 3차원 그래픽 라이브러리를 개발하여 사용하여 왔으며, 이를 공개용으로 배포한 것이 OpenGL이다. OpenGL은 아주 훌륭한 3차원 그래픽 라이브러리로서 카드, 게임, 애니메이션, 영상처리 등에 많이 사용되고 있으며, 운영체제에 관계없이 Windows나 Linux O/S에서 사용 가능하다.

프로그래밍 언어로는 객체 지향 언어인 C++을 사용하도록 하였으며, C++언어의 컴파일러는 Linux 운영체제에 포함되어 있는 gcc 컴파일러를 사용하도록 한다. Windows에서는 마이크로소프트 윈도우즈로 포팅(Porting)된 GNU 소프트웨어 개발 도구인 Mingw (Minimalist GNU for Windows)를 사용하면 gcc 컴파일러를 사용할 수 있다.

2.2 전처리장치의 구성

본 연구에서 개발을 진행하고 있는 전처리장치를 구성하는 기능모듈을 다음에 나열해 보았다.

- 키보드 명령어 입력 처리 모듈
 - 메뉴 선택 처리 모듈
 - 3차원 그래픽화면 디스플레이 모듈
 - 격자 작성 및 수정 모듈
 - 경계조건 작성 및 수정 모듈
 - 유체의 물성치 데이터베이스 모듈
 - 기타 제어변수 설정 모듈
 - template 모듈
 - 다른 프로그램과의 데이터 호환을 위한 데이터 변환 모듈
- 물론 위에 나열한 모듈들이 프로그램의 전부는 아니며, 계속

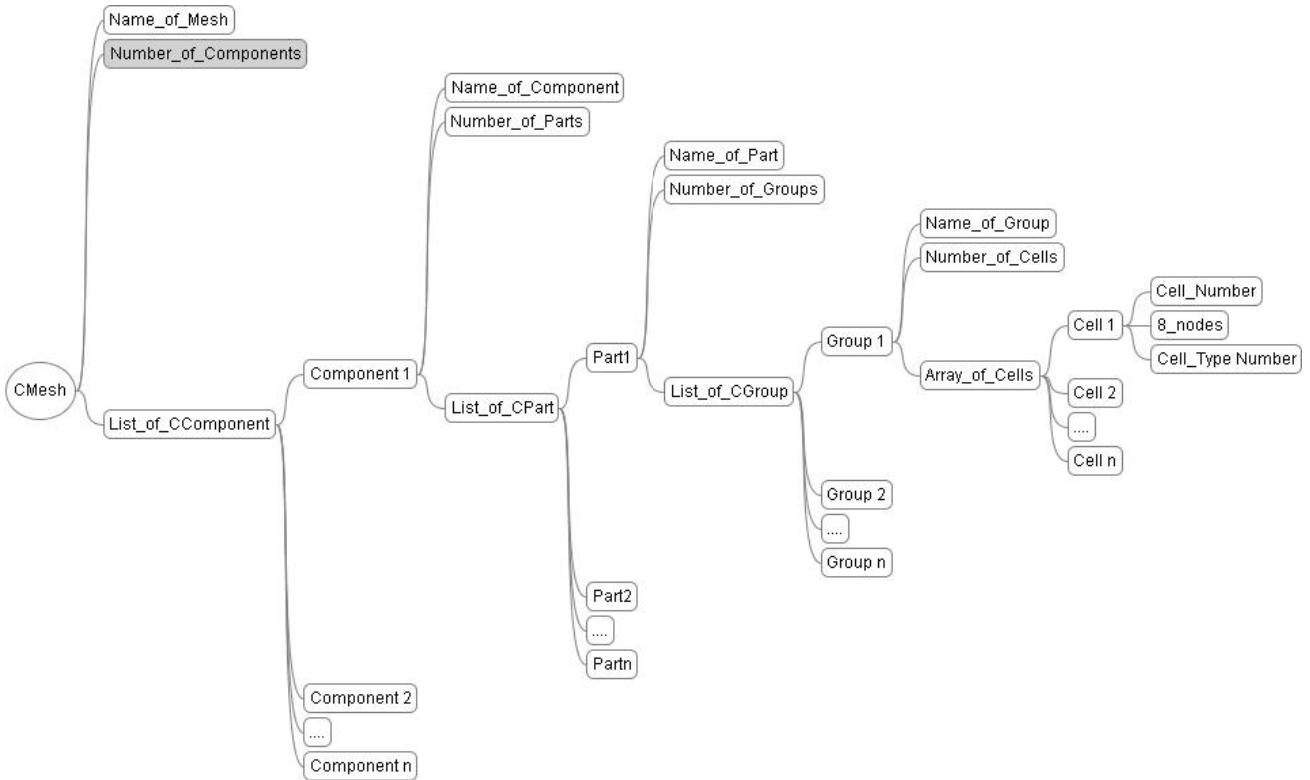


Fig. 2 Memory structure

하여 많은 부분이 추가로 더해져야 할 것이다.

3. 개발 결과

이렇게 선정한 wxWidgets GUI라이브러리와 gcc 컴파일러를 사용하기 위하여, 프로그램을 제작하는 통합개발환경(Integrated Development Environment, IDE)으로서 wxDev-Cpp[21]라는 프로그램을 사용하도록 한다. 이 wxDev-Cpp는 Dev-Cpp[22]이라는 IDE 프로그램에 wxWidgets 라이브러리를 사용하기 쉽게 내장시킨 프로그램이다. 이름이 생소하다고 느끼는 독자들은 마이크로소프트사의 Visual Studio와 같은 기능을 하는 프로그램이라고 생각하면 된다.

Fig. 1에 나타낸 것과 같이, wxDev-C++를 이용하여 MDI(Multi Document Interface) 구조의 Frame을 구성하여 프로그램을 작성하기 시작하였다. MDI구조란 서로 다른 여러 개의 데이터파일을 동시에 열고 작업할 수 있는 구조이다. 일반적인 워드프로세서에 많이 사용되는 구조이며, 한 데이터 파일에서 다른 데이터 파일로 자료를 복사하거나 주 데이터 파일을 비교하기가 쉬운 장점이 있다. 여기에서도, 이미 만들어진 메쉬를 다른 파일에 복사한다거나, 두 개의 메쉬파일을 서로 비교한다거나 할 때의 편의를 위해 MDI 구조를 채택하였다.

먼저 메쉬를 구성하는 메모리 구조를 설계하는데 있어서,

비정렬 격자계를 채택하였고, 멀티 블록 메쉬의 사용이 많아질 것으로 예상하였으며, 멀티 프로세서를 이용한 병렬처리 또한 고려할 필요가 있다고 판단하였다. 따라서 메쉬를 만들 때 독립적인 여러 개의 메쉬로 구성된 복합적인 메쉬를 구성하기 쉽도록 하기 위하여 트리구조를 사용하기로 결정하였다. 메쉬는 여러 개의 콤포넌트(Component)로 구성되었고, 콤포넌트는 여러 개의 파트(Part)로 구성되었으며, 파트는 여러 개의 그룹(Group)으로 구성되었다. 각 그룹은 셀 어레이(Cell Array)를 가지고 있게 된다. 이렇게 메모리를 구성하면 복잡한 형상을 만들 때에 하나의 콤포넌트가 여러 개의 파트와 그룹으로 구성될 수 있으므로, 혼자 작업을 하거나 여러 명이 나누어서 협업을 하기에 수월할 것으로 생각한다. 또한, 병렬처리를 할 때 블록을 나누어 설정하기에도 편리한 장점이 있게 된다. Fig. 2에 메모리 구조를 나타내었다. 그림은 FreeMind[22]라는 프로그램을 사용하여 작성하였다. FreeMind는 아이디어를 정리하기 위하여 사용하는 MindMap이라는 방법을 컴퓨터 프로그램으로 만든 것인데, 무료로 사용할 수 있다. 트리 구조를 그림으로 나타낼 필요가 있을 때 사용하면 유용하다.

Fig. 3에 wxWidgets GUI Library를 사용하여 만든 기본 프레임워크를 나타내었다. 프로그램의 이름을 일단 "MPP (Multi-Platform Pre-processor)"라고 지었다. 윈도우 운영체제에서 실행되는 대부분의 프로그램과 유사한 형태를 가지는 것을 볼

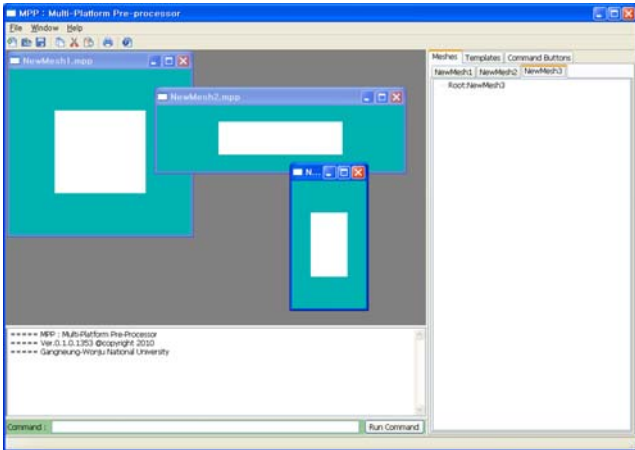


Fig. 3 Main frame

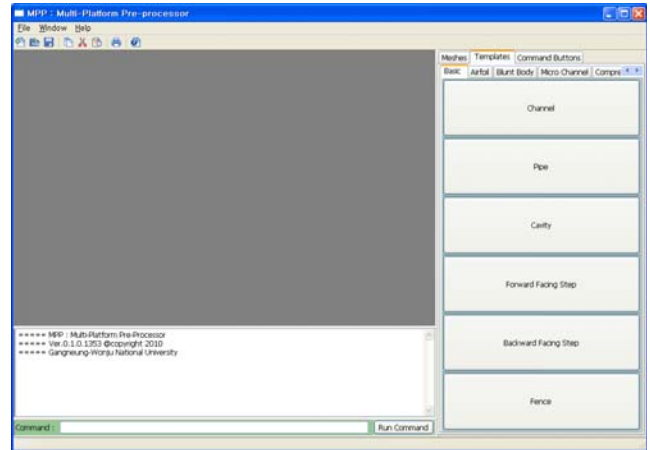


Fig. 5 Template menu buttons

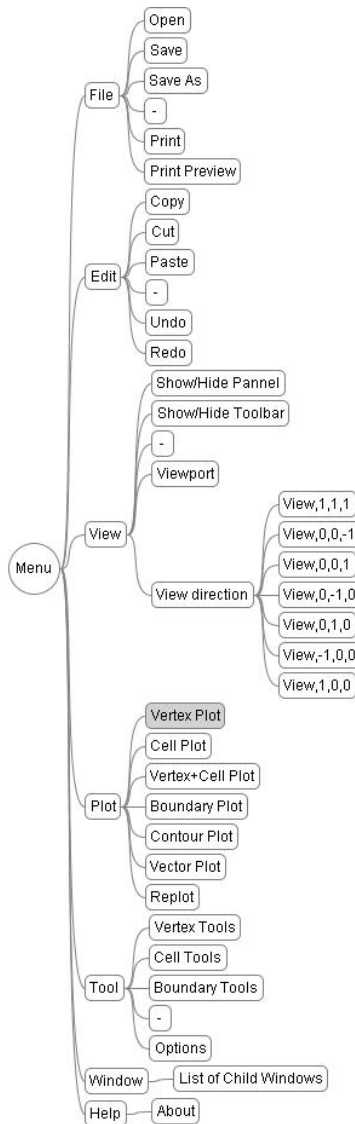


Fig. 4 Menu

수 있다.

왼쪽 위의 회색부분에는 Child Window를 사용하여 여러 개의 화면을 나타낼 수 있는 Display 부분이며, 현재는 기본적인 하얀색 사각형이 나타나 있는 샘플윈도우를 여러 개 보여 주고 있다. 오른쪽에 분할된 화면은 메쉬 모듈, 템플레이트 모듈, 명령어 모듈 등의 입력도구를 위한 여러 개의 윈도우를 노트북 위젯(Widget)을 이용하여 나타내고 있다. 왼쪽 중간부분은 명령을 수행한 결과를 텍스트로 나타내는 출력화면이다. 왼쪽 맨 아래의 창은 키보드 입력을 받아들여 명령을 수행하는 명령어 입력 윈도우이다.

Fig. 4에는 메뉴 트리의 설계를 나타내었다. 3차원 캐드 프로그램들이 가지고 있는 메뉴와 비슷한 것들을 준비하고 있으며, 아직 프로그램에는 File, Window, Help의 세 개의 기본적인 메뉴만 삽입되어 있지만, 향후 계속하여 필요한 메뉴 기능을 추가해 나갈 예정이다.

Fig. 5에는 템플레이트 메뉴 버튼 화면을 나타내었다. 학교에서 CFD를 교육할 때 많이 사용하는 간단한 유동을 해석하기 위하여 메쉬를 작성할 때, 사용자가 쉽게 사용할 수 있도록 미리 대부분의 기능을 하나의 화면에서 입력할 수 있도록 만든 것이다. 우선적으로 사각형 채널, 원형 파이프, 캐비티, 후향계단 유동 등 기본적으로 테스트에 많이 쓰이는 격자를 쉽게 작성할 수 있도록 템플레이트를 구성하였다.

Fig. 6에는 3차원 사각형 채널 유동해석을 위한 격자를 작성할 수 있는 템플레이트 화면을 나타내었다. 화면의 왼쪽 위에는 해석할 유동의 영역을 그림으로 표시하여 이해를 쉽게 하였고, 각각의 치수를 기호로 나타내었다. 화면의 아래쪽 공간에 각각의 치수를 입력하는 입력부를 만들었으며, 그림의 기호에 대한 각각의 좌표에 대한 시작과 끝의 수치, 각 방향으로의 격자의 개수, 격자간의 비율(등비수열의 공비)을 입력하고 “OK” 버튼을 누르면 격자가 만들어지게 된다. x, y, z 각 방향으로 등비수열을 이용하여 격자의 간격을 조절할 수

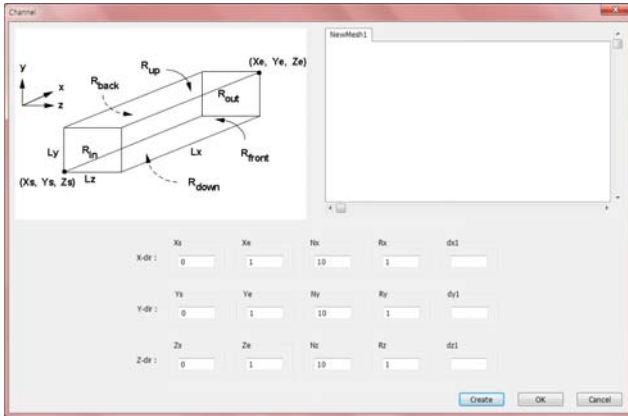


Fig. 6 Dialog window for generating mesh for channel flow (cartesian coordinate)

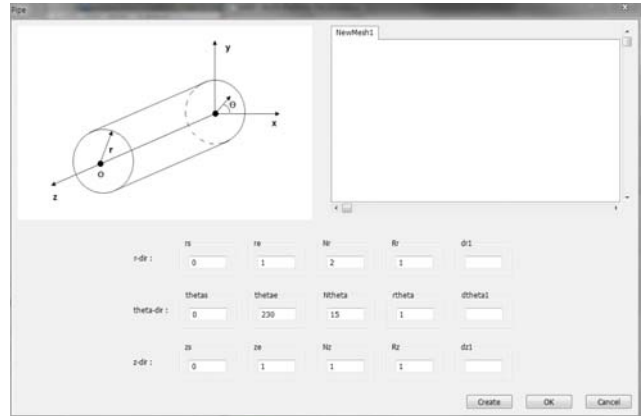


Fig. 8 Dialog window for pipe flow (cylindrical coordinate)

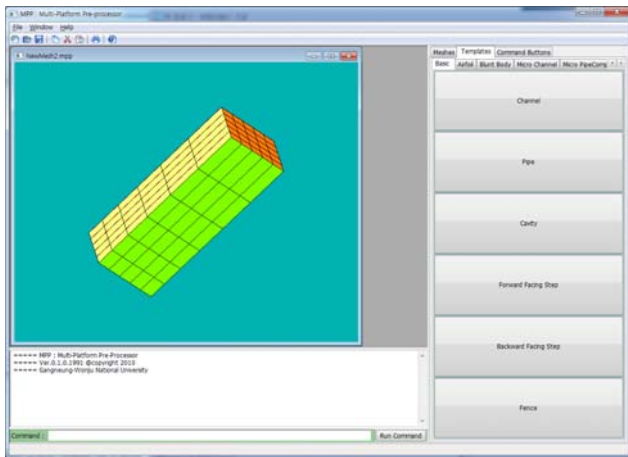


Fig. 7 Created mesh for channel flow

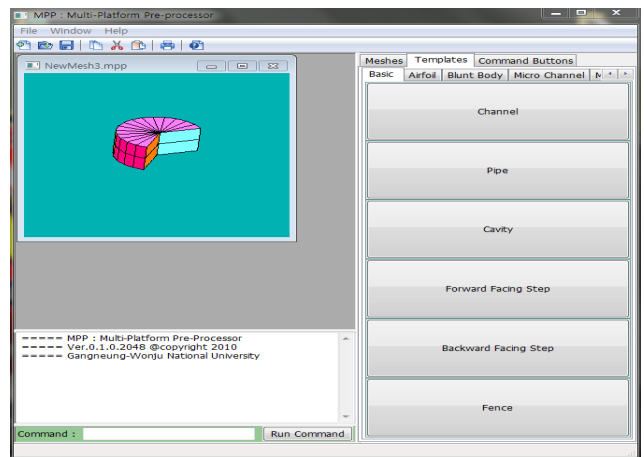


Fig. 9 Created mesh for pipe flow

있도록 입력 변수를 설계하였다. 오른쪽 윗부분은 메시의 트리구조를 나타내기 위한 창이다.

Fig. 7에는 템플레이트에서 만든 3차원 채널 격자를 나타내었다. 길이방향의 격자는 비율을 주어 만들었기 때문에 격자 간격이 일정하지 않고 오른쪽으로 가면서 점점 커지는 것을 볼 수 있다. 쉽게 알아보기 위해 각 면의 색상을 다르게 나타내었다. Fig. 8에는 원형 파이프 격자생성 템플레이트 화면을 나타내었다. 각각 r , θ , z 방향으로 시작과 끝의 반지름, 시작과 끝의 각도, 시작과 끝의 z 좌표 값을 입력하고 “OK” 버튼을 누르면 격자가 생성된다. Fig. 9에는 이 원형 파이프 격자생성 템플레이트를 이용하여 만든 간단한 격자이다. 시작 각도와 끝 각도를 적절히 지정하면 그림과 같은 케이크 모양의 격자도 만들 수 있다. 0이 아닌 시작 반지름 값을 지정하면 동심원관을 만들 수도 있다.본 프로그램은 이렇게 템플레이트의 기능을 위주로 메시작성을 하는 것이 특징으로 하며, 향후 계속 템플레이트를 추가할 예정이다.

4. 결 론

어느 한 회사의 시스템에 종속되지 않도록, 윈도우즈와 리눅스에서 모두 사용이 가능한 CFD 전처리장치를 개발하였다. 주로 학교에서 학생들이 쉽게 배울 수 있도록, 여러 가지 형상에 대한 유동격자를 쉽게 생성할 수 있는 템플레이트를 제공하는 것을 특징으로 한다.

향후 여러 가지 형상의 격자에 대한 템플레이트를 계속하여 추가할 예정이며, CFD초보자들의 실습 등에 많은 도움이 되기를 기대한다.

후 기

이 논문은 강릉원주대학교 2009년도 장기해외파견연구 지원에 의하여 수행되었습니다.

Note

This paper is a revised version of a paper presented at the KSCFE 2013 Spring Annual meeting, Seogwipo, Jeju, May 15-16, 2013.

References

- [1] 2002, Hur, N., Kim, S.R., Won, C.S. and Hur, S., "Development of a General Purpose Program for 3-D Flows," *Proceedings of the KAMES 2002 Joint Symposium A, COEX Conference Center (Nov. 13, 2002), Seoul*, pp.1984-1989.
- [2] 2004, Hur, N., *Development of General Purpose Thermo/Fluid Flow Analysis Program*, MI-0129-00-0001, Sogang University.
- [3] 2007, Kim, S.R., Yeo, J.-H., Won, C.S. and Hur, N., "Development of Pre/Post Processor Program for NUFLEX," *Proceeding of the KSCFE Spring Annual Meeting*, pp.91-94.
- [4] <http://www.vtk.org>
- [5] 2006, Myong, H.K. and Ahn, J.K., "Development of a Pre-Processing Program for Flow Analysis," *Proceedings of The Fourth National Congress on Fluids Engineering, Kyungju (August 23-25, 2006), Korea*, pp.897-900.
- [6] 2006, Myong, H.K. and Ahn, J.K., "Development of a Post-Processing Program for Flow Analysis," *Proceedings of The Fourth National Congress on Fluids Engineering, Kyungju (August 23-25, 2006), Korea*, pp.901-904.
- [7] 2008, Myong, H.K. and Ahn, J.K., "Development of a Pre-Processing Program for Flow Analysis Based on the Object-Oriented Programming Concept," *Trans. of KSME B*, Vol.32, No.1, pp.70-77.
- [8] 2008, Myong, H.K. and Ahn, J.K., "Development of a Post-Processing Program for Flow Analysis Based on the Object-Oriented Programming Concept," *Trans. of KSME B*, Vol.32, No.1, pp.62-69.
- [9] 2009, Lee, J.H., Cho, K.W. and Kim, B.S., "Development of 2-D Unstructured Hybrid Grid Generation Program using Java Applet," *Proceeding of the KSCFE Autumn Annual Meeting*, pp.65-70.
- [10] 2010, Lee, J.H., Cho, H.S. and Kim, B.S., "Development and Application of Automatic Grid Generation Program for 3-D wing using Java Applet," *Proceeding of the KSCFE Spring Annual Meeting*, pp.335-340.
- [11] 2011, Song, J.H., Kang, S.H., Kim, G.M. and Kim, B.S., "A Study of the application of Delaunay Grid Generation on GPU using CUDA Library," *Proceeding of the KSCFE Spring Annual Meeting*, pp.194-198.
- [12] <http://www.msharp CFD.com>
- [13] 2011, Kim, S.R., "The Development of Multi-platform CFD Pre-processor," *Proceeding of KSME Fall Annual Meeting, EXCO (Daegu)*, p.426.
- [14] 2012, Kim, S.R., "The Development of Multi-platform CFD Pre-processor(II)," *Proceeding of KSME Fall Annual Meeting, CECO (Changwon)*, p.228.
- [15] 2006, Smart, J., Hock, K. and Csomor, S., *Cross-Platform GUI Programming with wxWidgets*, Prentice-Hall.
- [16] <http://www.wxwidgets.org>
- [17] 1993, OpenGL ARB, *OpenGL Programming Guide*, Addison Wesley.
- [18] 1997, OpenGL ARB, *OpenGL Reference Manual*, 2nd ed., Addison Wesley Developers Press.
- [19] 1996, Wright, R.S. Jr. and Sweet, M., *OpenGL Superbible*, Waite Group Press.
- [20] 1997, Fosner, R., *OpenGL Programming for Windows 95 and Windows NT*, Guide.
- [21] <http://wxdsgn.sourceforge.net/>
- [22] <http://www.bloodshed.net/dev/devcpp.html>
- [23] http://freemind.sourceforge.net/wiki/index.php/Main_Page