

관내 유동 해석을 위한 웹기반 자바 프로그램 개발

김 경 성,¹ 박 종 천^{*2}

DEVELOPMENT OF WEB-BASED JAVA PROGRAM FOR NUMERICAL ANALYSIS OF PIPE FLOW

K.S. Kim¹ and J.C. Park^{*2}

In general, flow analysis programs have been built in not JAVA language but such as Fortran, C or C++ because of a merit in performance of computation. However, most developer of those languages are not free for use. User should purchase the development tools of programs, and install it on their own computer. Fortunately, JAVA is a free software, and can be used to develop WEB-based programs. It is well known that the JAVA language is time consuming for calculating. However, the performance of computer power has been increased continuously, so it is sure that the former states can be overcome. In the present study, a flow analysis program, which is interactive with Graphics, using the Web-based programing language, JAVA, has been developed for solving the two-dimensional flow motion.

Key Words : 유동해석(Flow Analysis), 웹기반 자바 언어(Web-based JAVA Language), 관내유동(Pipe Flow), 나비에-스톡스 방정식(Navier-Stokes Equation)

1. 서 론

개인용 컴퓨터의 발달과 수치해석 기법의 발달로 인해 공학적 문제를 해결함에 있어서 실험과 함께 전산 유체 역학(이하 CFD)이 함께 사용되어 왔다.

이전까지의 CFD는 포트란(Fortran), C, C++ 의 프로그래밍 언어와 매트랩(Matlab)과 같은 수치해석적 도구(tool), 그리고 플루언트(Fluent) 등과 같은 CFD 상용 도구에 의해 진행되어져 왔다. 포트란은 연산속도가 빠르다는 장점을 가지며, C나 C++은 객체 지향 언어를 사용하여 프로그램 구성에 있어서 편리함과 그래픽 처리가 포트란에 비해 강하다는 장점을 가지고 있다. 또한, CFD 상용 도구들은 직접 프로그램을 작성하지 않고, 사용자에게 간단히 경계조건을 수정 후 수치해석을 실행할 수 있다는 장점이 있다. 그러나 이러한 장점들에 비해 각각 언어에 대한 컴파일러(compiler)와 빌더(builder)가

없으면 사용이 불가능하고, 이와 관련된 프로그램을 구입하여 개인용 컴퓨터 또는 연산을 수행하고자 하는 컴퓨터에 설치 후 수치해석을 실행해야 하는 제약이 따르게 된다.

이에 대해, 자바는 컴파일 프로그램과 빌더가 무료로 공개되어 있어, 웹(web)에 접속할 수 있다면 시간적·공간적 제약 없이 다운로드하여 프로그램의 개발 및 실행 할 수 있다. 또한 완성된 프로그램이 있을 시에는 완전한 버전(full version)의 컴파일러와 빌더가 아닌 실행 환경 프로그램(runtime environment program)만 설치 한 후 간단히 수치해석을 실행할 수도 있다. 자바는 웹상에서 사용될 수 있는 객체지향 언어이며, 수치해석 프로그램을 외부기억장치 등의 도움 없이 웹상에서 즉시 적용·실행하여 사용자 의견이 반영된 프로그램 개발을 비교적 용이하게 할 수 있다. 또한, 자바는 연산뿐만 아니라 그래픽에 관련되어서도 많이 발달이 되어 있으며, 이를 이용하여 도구를 만들 수 있다는 장점이 있으며, 이는 C, C++ 그리고 Visual C등과 동일한 기능이기는 하지만 웹에서 사용되어 질 수 있다는 자바만의 특성이 있다. 한편, 단점으로는 연산속도가 상대적으로 느려 수치해석용으로는 부적합 하다는 평이 지배적이었으나 이는 최근 지속적인 컴퓨터 성능의 발달로 충분히 극복될 수 있으며, 상술한 장점들로 인하여 보완이 가능하다.

접수일: 2010년 2월 9일, 수정일: 2010년 3월 5일,
계재확정일: 2010년 3월 12일.

1 정회원, Civil Engineering, Texas A&M Univ.

2 정회원, 부산대학교 조선해양공학과

* Corresponding author, E-mail: jcpark@pusan.ac.kr

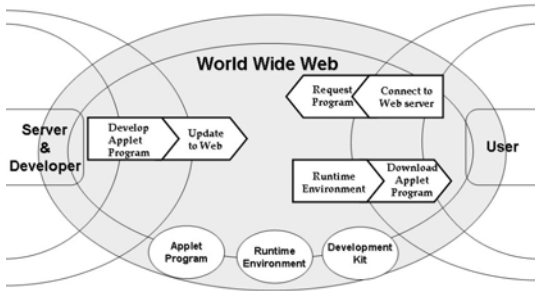


Fig. 1 How to develop and use applet program

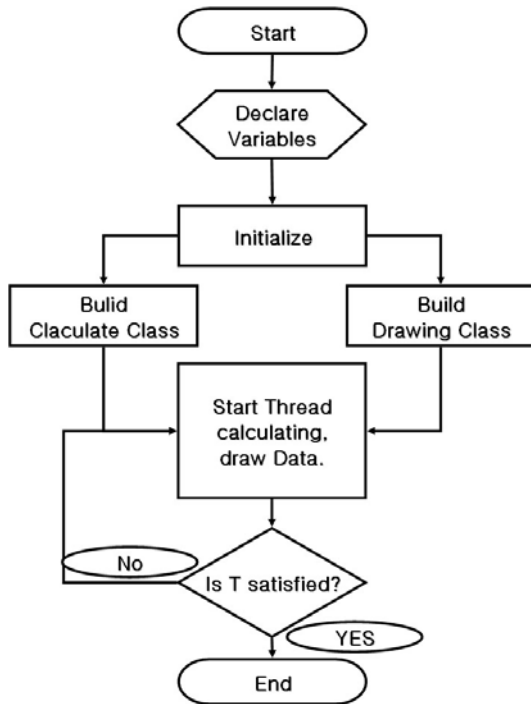


Fig. 2 Algorithm for JAVA program

본 연구에서는 객체지향의 웹기반 언어인 자바를 이용하여, 단순한 대류 방정식의 이산화 과정 중 공간 이산화에 각종 수치도식(Scheme)을 적용하고, 또한 2차원 관내 유동을 해석하는 프로그램을 개발한다. 양자 모두 시뮬레이션 결과를 후처리 하는 그래픽 프로그램과 연동되어 개발이 된다. 특히 계산 성능면에서 기존의 포트란 언어로 구사된 프로그램과 비교하여 자바를 이용한 수치계산 기법 개발의 가능성을 살펴해보았다. 개발된 프로그램은 웹에서 사용가능한 애플릿(Applet) 프로그램으로 작성되어 향후 웹기반의 유동 해석을 위한 기반을 마련하기로 한다.

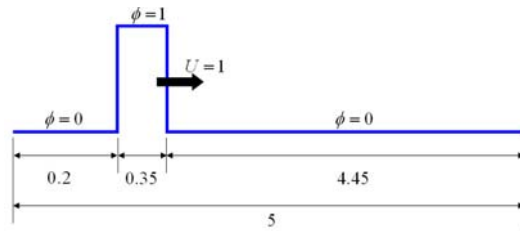


Fig. 3 Test problem for 1-D wave propagation

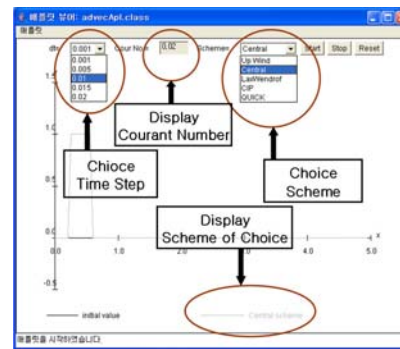


Fig. 4 Menu selection in the present JAVA program

2. JAVA 프로그램의 구성

2.1 애플릿(Applet)

애플릿이란 Little Java Application의 줄임말로 자바 언어로 작성된 소프트웨어 중 별도의 애플릿 형식을 뜻한다. 크기가 작아서 네트워크에서의 전송에 적합하고, 웹상에서 배포가 가능하다. 웹에서 사용하는 표준 데이터 형식인 하이퍼텍스트 생성 언어(HTML)로 작성한 문서에 애플릿이라는 태그를 사용하여 자바 애플릿을 지정한다. 자바 애플릿을 동작시키는 데는 자바 가상 머신을 내장한 웹 브라우저가 필요하다. 브라우저는 불러내 온 문서 속에 애플릿이라는 태그가 있으면, 지정된 자바 애플릿을 웹 서버로부터 다운로드하여 실행한다 (Fig. 1).

2.2 데이터 처리 방식

Fig. 2는 수치해석 및 데이터 처리 과정을 보여준다. 먼저 연산 클래스에서 연산을 한 뒤, 데이터 클래스에서 각 데이터를 수집 및 분류를 한다. 그 뒤 데이터 클래스에서 수집 및 분류되어진 값과 그리기 클래스의 설정을 스레드(Thread)가 참조하여 수집 및 분류되어진 데이터를 그리기를 한다. 이 일련의 과정을 애플릿에서 구동하게 하여 웹과 연동되어지는 수치해석 프로그램이 만들어 진다.

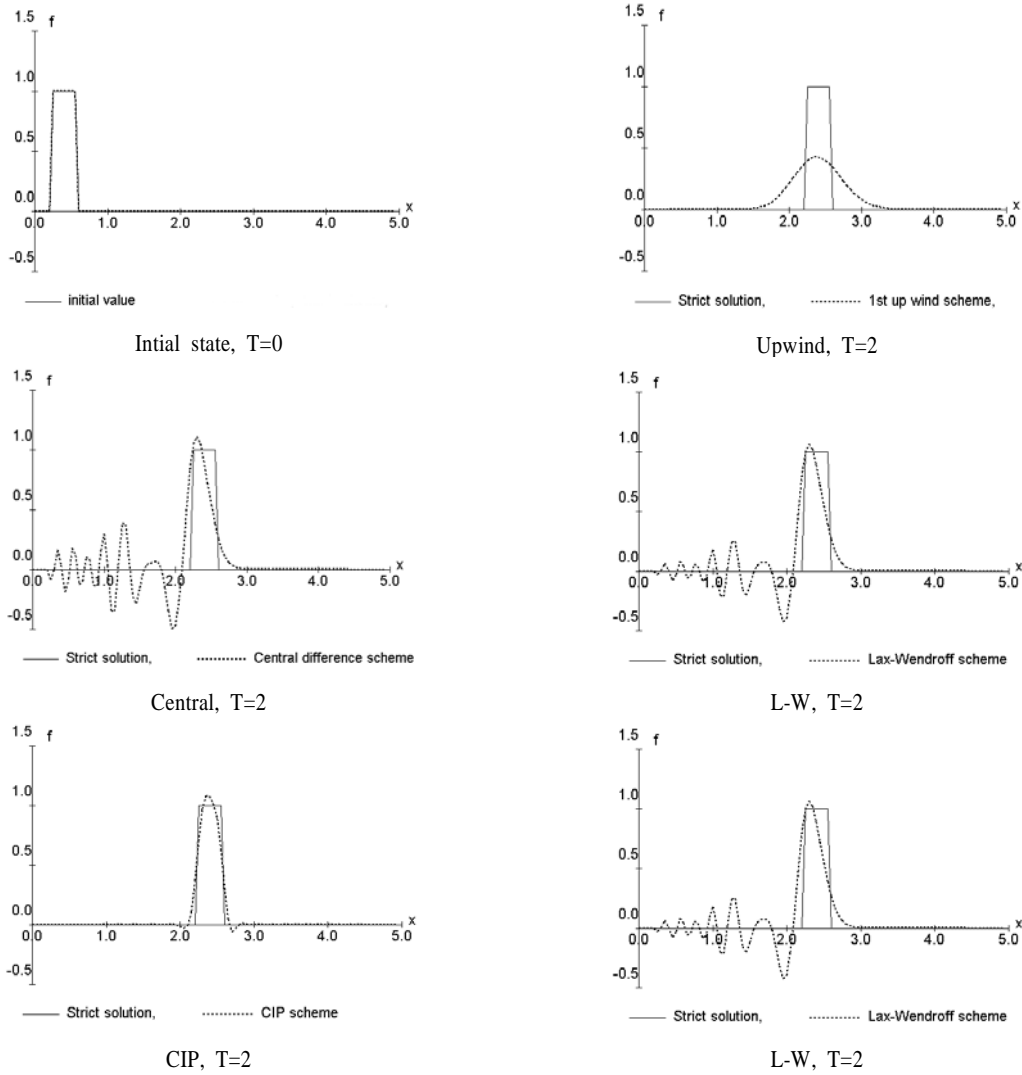


Fig. 5 Wave configuration obtained applying various numerical scheme at $\Delta t=0.001$ and $\Delta x=0.05$

3. 1차원 파동 방정식의 수치 시뮬레이션

3.1 문제 설정

1차원의 테스트 문제로는 물리량 ϕ 의 1차원 파동(대류) 방정식의 보존형인 식 (1)의 수치 시뮬레이션을 택하였다.

$$\frac{\partial \phi}{\partial t} = -U \frac{\partial \phi}{\partial x} \quad (1)$$

단, t 는 시간, U 는 x -방향에서의 이류속도를 나타낸다.

식 (1)을 물리량 ϕ_i 를 격자중심에 배치하는 정규격자를 채용하여 유한체적법에 의해 이산화하고, 시간적분에는 1차 정확도의 오일러(Euler) 도식을 적용하면,

$$\phi_i^{n+1} = \phi_i^n - \Delta t \frac{F_{i+1/2}^n - F_{i-1/2}^n}{V_i} \quad (2)$$

$$F_{i+1/2}^n = u_{i+1/2}^n \phi_{i+1/2}^n S_{i+1/2} \quad (3)$$

여기서 V_i 는 격자 i 의 검사체적, $S_{i+1/2}$ 은 격자경계 $i+1/2$

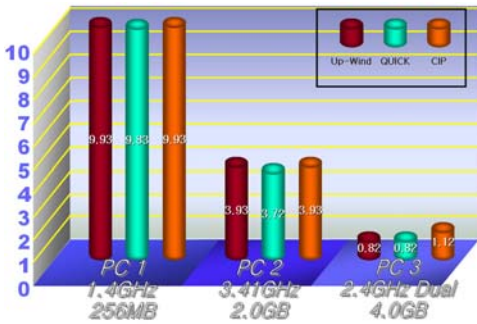


Fig. 6 Comparison of CPU time

의 면적, $F_{i+1/2}$ 은 격자경계 $i+1/2$ 의 운동량 유속(Flux), 위 첨자 n 은 시간스텝을 각각 나타낸다. 수치도식으로는 1차 상류차분(1st-upwind scheme)과 중심차분, CIP(Cubic-interpolation pseudo- particle), QUICK(Quadratic upstream interpolation for convective kinematics), 그리고 Lax-Wendroff(이하, L-W) 도식의 5종류를 선택하도록 프로그램을 구성하였다.

3.2 시뮬레이션 조건

식 (1)의 함수 ϕ 는 Fig. 3에 보이는 바와 같이 계단함수로 정의되며, ϕ 의 초기 값으로는 전체 $x=5$ 의 길이 중 $x=0.2\sim 0.55$ 의 영역에 $\phi=1$ 의 값을, 나머지 영역에서는 $\phi=0$ 의 값을 주었다.

Fig. 4에서 보이는 바와 같이, 본 프로그램에서는 시간간격과 수치도식을 사용자가 선택할 수 있으며, Δx 의 값은 프로그램 내에서 지정된 0.05를 쓰도록 되어 있다. 한편 변화된 시간간격 Δt 에 따라 Courant 수는 자동적으로 표시되도록 하였다. 또한 수치도식을 선택하면 좌측 하단에 초기 값과는 다른 색으로 선택된 수치도식이 표시된다. 좌측 상단에는 "Start", "Stop", "Reset"의 세 개의 버튼을 선택할 수 있고, "Start"는 연산 및 그래픽 시작을, "Stop"은 연산 및 그래픽 멈춤을, 그리고 "Reset"은 초기 형태의 복원을 각각 실행하는 버튼이다. 연산 및 그래픽은 실시간으로 데이터를 받아서 매 시간간격마다 계산된 값을 그래픽 처리하도록 되어 있다.

3.3 시뮬레이션 결과

Fig. 5는 시간 간격 $\Delta t=0.001$ 및 격자간격 $\Delta x=0.05$ 의 조건 하에서 계단 함수인 ϕ 가 시간적으로 전파하는 과정의 공간적 분포를 나타낸다. 이 때 수치 도식 별로 웹상에서 실행한 화면을 캡처하여 나열한다. 상류차분 도식의 경우에는 심한 수치적 확산과 소산에 의해 원래 형태의 함수를 알아보기가 힘들고, 중심차분 도식에서는 심한 수치적 진동을 보이며, 고차정확도의 차분 도식에서는 대체로 안정적인 결과를 나타

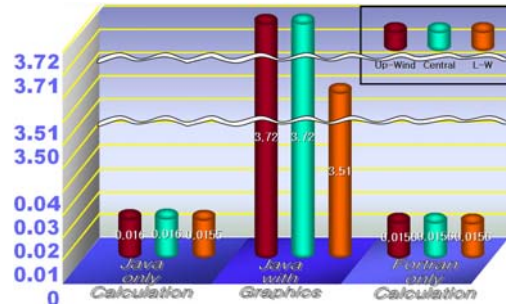


Fig. 7 Comparison of CPU time in PC2

내는 등 이전의 선행된 결과[5]와 유사한 결과를 보임을 알 수 있다.

3.4 연산속도의 검토

계산에 따른 CPU 시간을 비교하기 위해, 제안된 수치도식들 중 1차 상류차분, QUICK 그리고 CIP 도식에 대해서 Table 1에 나타낸 것과 같이 사양(Spec)이 각기 다른 세 대의 컴퓨터를 사용하여 프로그램을 실행하였다. 단, 무차원 시간이 $t=2.0$ 일 때까지 수행한 결과를 상호 비교하였다. Fig. 6은 컴퓨터의 사양에 따른 수치 도식별 연산 시간의 비교 결과를 나타낸다. 동일한 수치도식과 시간간격을 설정하더라도 컴퓨터의 사양에 따라 연산 시간에는 큰 차이가 있음을 확인할 수 있다.

또한, 제안된 수치 도식들 중 1차 상류차분, QUICK 그리고 CIP 도식에 대해 PC2를 사용하여 포트란 언어와 자바 언어와의 연산 시간을 비교하여 보았다. 이 때, 자바 언어는 그래픽 기능이 포함된 경우와 그렇지 않은 경우를 분리하여 비교하였다. Fig. 7에서 보이는 것과 같이 그래픽 기능이 포함되지 않은 단순 연산에 대해서는 포트란과 자바 사이에 그다지 큰 차이를 찾아볼 수 없었다. 한편, 그래픽 처리를 포함하는 경우 자바가 현저히 늦는 경향을 보였으나, 복잡한 그래픽 처리를 동시에 수행한다는 관점에서는 타당한 결과라고 볼 수 있다. 이때 자바 프로그램에서 그래픽 처리의 점유율은 전체 연산 시간의 약 96%에 해당한다.

Table 1 Spec of computers used in benchmark tests

1	PC1	<ul style="list-style-type: none"> ● CPU : Intel Pentium® M Process 1.4GHz ● RAM : 256MB ● VGA : Nvidia GeForce4 440 Go
2	PC2	<ul style="list-style-type: none"> ● CPU : Intel Pentium® 4 Process 3.41GHz ● RAM : 2.0GB ● VGA : Nvidia GeForce 6600
3	PC3	<ul style="list-style-type: none"> ● CPU : Intel Core™2 CPU 6600 @ 2.4GHz ● RAM : 4.0GB ● VGA : Nvidia GeForce 7300 LE

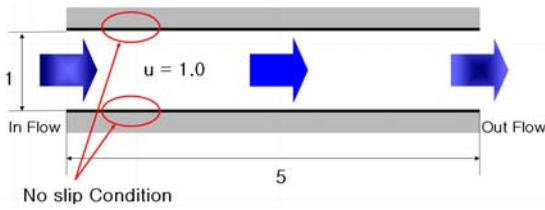


Fig. 8 Test problem for 2-D pipe flow

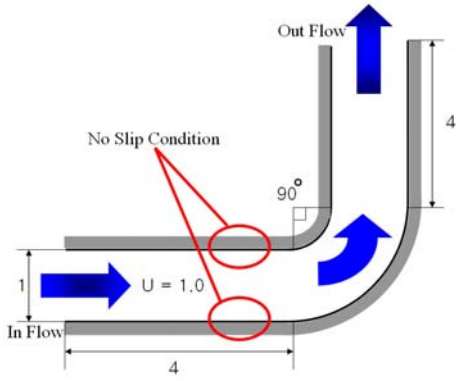


Fig. 9 Test problem for 2-D bended pipe flow

4. 1차원 관내 유동의 수치 시뮬레이션

4.1 문제 설정

2차원 관내 유동 해석에 관하여, 먼저 Fig. 8와 같이 직관 내의 유동을 고려하였다. 관의 한쪽을 유입, 반대쪽을 유출로 설정하였으며, 벽면은 점착 조건을 사용하였다. 다음으로 Fig. 9와 같이 90°로 굽은 관내 유동을 고려하였고, 모든 조건은 직관내 유동 해석과 동일하다.

지배방정식은 다음과 같이 연속방정식과 Navier-Stokes(이하 N-S) 방정식이다.

$$\nabla \cdot \vec{v} = 0 \tag{4}$$

$$\frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla) \vec{v} = -\nabla p + \frac{1}{Re} \nabla^2 \vec{v} \tag{5}$$

지배방정식은 대류항에 QUICK 도식과 Fractional Step(이하 FS)법을 이용하여 이산화된다. 일반적으로 FS법은 다음의 3단계를 거친다.

제 1단계로, 시각 n 에서 N-S방정식 (5)의 압력구배항을 생략하여 다음 식을 얻는다.

$$\frac{\vec{v} - \vec{v}^n}{\Delta t} = -(\vec{v}^n \cdot \nabla) \vec{v}^n + \frac{1}{Re} \nabla^2 \vec{v}^n \tag{6}$$

식 (6)에서 \vec{v}^n 이 기지의 값으로 주어지면, 중간 속도인 \vec{v} 를 구할 수 있다.

제 2단계에서는 시각 $(n+1)$ 에서의 속도 \vec{v}^{n+1} 는 식 (7)에 의해 압력구배항의 영향을 음적으로 보정해 준다.

$$\vec{v}^{n+1} = \vec{v}^n - \Delta t \Delta p^{n+1} \tag{7}$$

여기서 \vec{v}^{n+1} 이 연속방정식을 만족하도록 양변에 발산을 취한다.

$$\nabla \cdot \vec{v}^{n+1} = \nabla \cdot \vec{v} - \Delta t \nabla^2 p^{n+1} = 0 \tag{8}$$

이 식으로부터 얻어지는 압력의 Poisson 방정식

$$\nabla^2 p^{n+1} = \frac{\nabla \cdot \vec{v}}{\Delta t} \tag{9}$$

을 풀어 시각 $(n+1)$ 에서의 압력 p^{n+1} 를 구한다.

시각 $(n+1)$ 에서의 압력인 p^{n+1} 을 구하기 위해서는 압력구배항의 음적보정이 필요하다. 압력구배항의 보정에는 SOR법을 사용하며, 그 식은 다음과 같다.

$$P_{ij}^{n+1} = (1-w)P_{ij}^n + w \left\{ \frac{1}{4}(P_{i+1,j}^n + P_{i-1,j}^n + P_{i,j+1}^n + P_{i,j-1}^n) + \frac{\Delta x^2}{4} g_{ij} \right\} \tag{10}$$

$$\begin{bmatrix} AB \\ BAB \\ BAB \\ BAB \\ BA \end{bmatrix} \begin{bmatrix} \cdot \\ \cdot \\ P_{ij} \\ \cdot \\ \cdot \end{bmatrix} = \begin{bmatrix} \cdot \\ \cdot \\ g_{ij} \\ \cdot \\ \cdot \end{bmatrix} \tag{11}$$

단 w 는 완화계수 g_{ij} 는 소오스 항, P_{ij} 는 3중대각행렬의 역행렬을 구하여 얻어질 수 있다.

FS법의 마지막 제 3단계에서는 다음 식을 이용하여 시각 $n+1$ 에서의 속도 \vec{v}^{n+1} 를 갱신한다.

$$\frac{\vec{v}^{n+1} - \vec{v}}{\Delta t} = -\Delta p^{n+1} \tag{12}$$

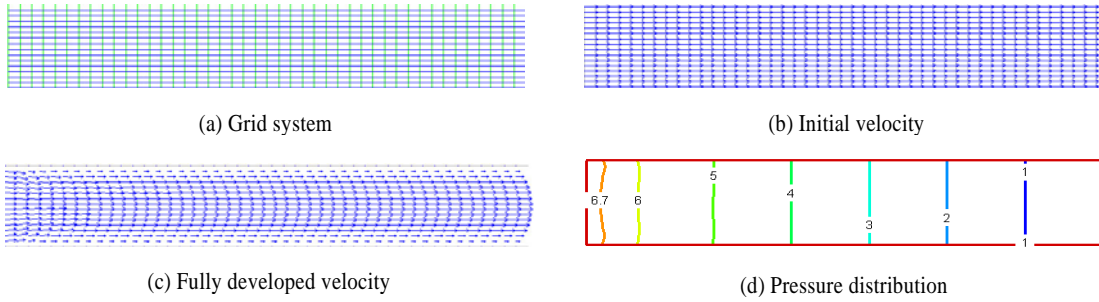


Fig. 10 2-D pipe flow analysis

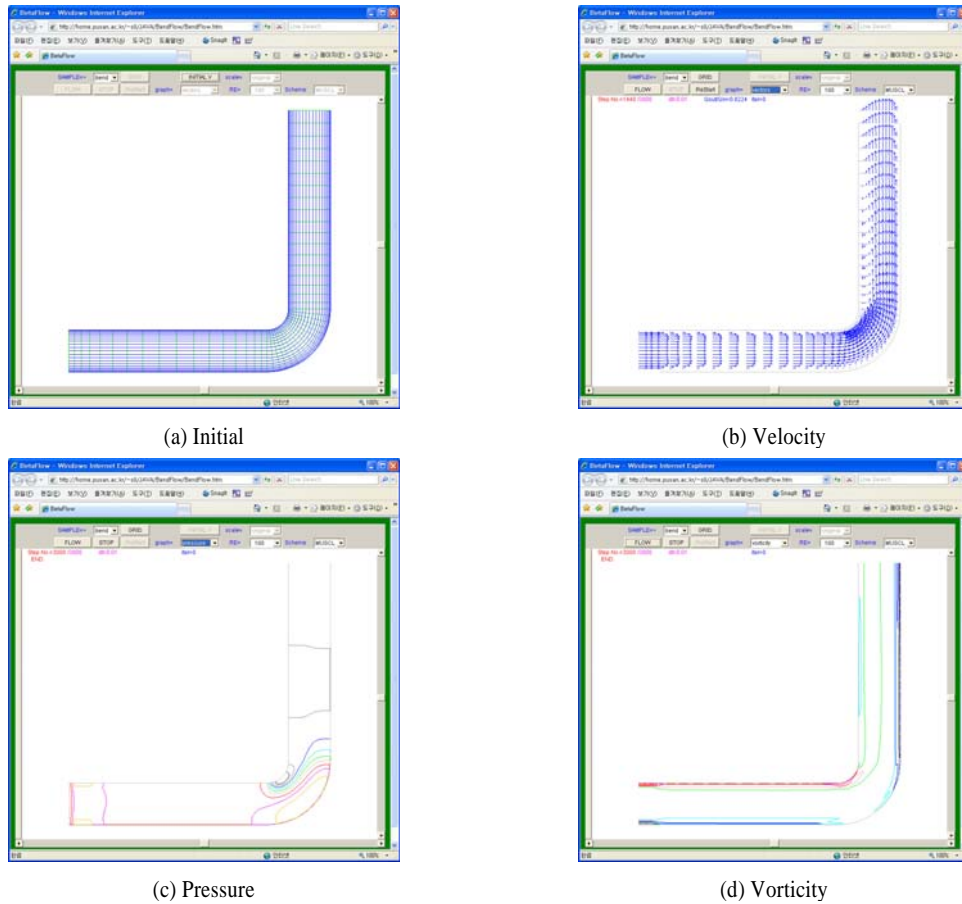


Fig. 11 2-D Bended flow analysis

4.2 시뮬레이션 결과

시뮬레이션 조건으로, 비압축성 유동의 $Re=100$ 인 층류로 가정하였다. Fig. 10(a)에 보이는 바와 같이 총 49×19 개의 등간격자를 사용하였으며, 시간간격은 $\Delta t=10^{-3}$ 으로 설정하였다. 초기 유동장에는 전 계산점에서 Fig. 10(b)에서와 같이 유동속도 $u=1.0$ 을 주었다.

Fig. 10(c)와 (d)에서 보이는 것과 같이 본 프로그램으로부터 계산된 결과는 내부의 그래픽 프로그램과 연동되어 실시간으로 웹상에서 가시화 될 수 있도록 작성되었다. 시간이 흐름에 따라 각 계산 점에서 계산된 속도와 압력의 등고선이 표현된다. 한편, 단순 직관이 아닌 2차원 굽은 관내 유동 해석 프로그램역시 같은 조건을 이용하여 해석하였다. 이 역시

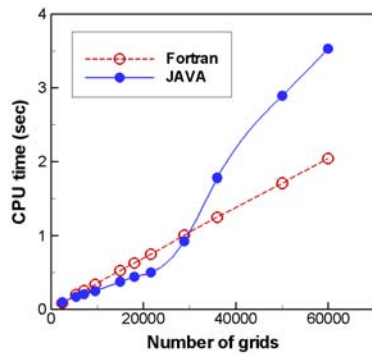


Fig. 12 Comparison of CPU time between FORTRAN and JAVA programs for 2-D pipe flow analysis

Fig. 11에 보이는 바와 같이 속도와 압력 및 와도(Vorticity) 분포가 내부의 그래픽 프로그램과 연동하여 실시간으로 가시화 되도록 구성되어 있다.

연산 속도의 비교를 위해 단순 직관의 경우 동일한 알고리즘을 사용하여 포트란언어와 자바언어로 작성된 프로그램을 실행시켜 격자수 증가에 따른 언어 간 연산속도를 비교하여 보았다. 이때, 자바의 경우 그래픽 관련 루틴을 제외하여 계산을 수행하였으며, 한 시간 스텝당 평균속도를 비교하였다. Fig. 12에 보이는 바와 같이, 격자수가 증가함에 따라 두 경우 모두 연산 시간이 증가함을 보이나, 증가하는 경향에 대해서는 다소 상이한 결과를 보였다. 포트란의 경우 격자수의 증가에 대해 선형적인 증가추세를 보이는 반면, 자바의 경우 격자수가 2만개정도 까지는 포트란에 비해 빠른 연산속도를 보이며 선형적으로 증가하지만 2만개 이후 연산시간이 비선형적으로 증가하고 포트란에 비해 연산속도가 느려진다. 이는 각 클래스를 독립적으로 인식하여 동일 시간에 다수의 연산을 가능하게 하는, 자바의 객체 지향적 특징으로 인해, 상대적으로 작은 격자수에서는 비교적 빠른 연산 속도를 보인다. 그러나 자바 또는 고급언어가 갖고 있는 느린 연산 속도에 의해 격자수가 증가함에 따라 연산 속도는 비선형적으로 증가하게 된다. 따라서 전체 연산은 객체지향성에 의해 자바가 빠를 수 있으나 각 클래스 내부 연산 시간의 증가로 인해 전체 연산 시간은 포트란에 비해 느려지기 때문이라고 판단된다. 자바의 객체 지향적 특징에 따른 결과라 생각되며, 향후 프로그래밍 언어의 성격에 적합한 프로그램 설계가 필요함을 알 수 있다.

5. 결 론

본 연구에서는, 자바를 이용하여 웹과 연동 가능하며 사용자의 설정이 가능한 애플릿 프로그램을 개발하였으며, 유동장

해석의 기본이 될 수 있는 비정상 이류 문제 및 2차원 관내 유동에 적용하였다. 수치해석된 결과는 자바의 그래픽 기능을 이용하여 후처리 기능을 연계하였다. 자바의 단점인 연산시간에 대하여, 각종 수치도식, 컴퓨터 사양, 그리고 프로그래밍 언어별 비교를 통하여 연산 시간의 차이를 확인 하였다. 그래픽 처리 과정이 배제된 경우, 자바는 격자수가 작을 때는 포트란에 비해 오히려 빠른 연산속도를 보여주었으나 격자수가 증가함에 따라 연산속도가 느려짐을 확인 할 수 있었다. 특히 격자수에 대해 자바는 비선형적인 연산시간의 증가를 보여주며, 이는 향후 자바의 객체 지향적 특징에 적합한 프로그램 설계가 필요함을 시사해준다. 향후 웹 기반의 3차원을 포함한 다양한 유동 해석 S/W의 개발 및 개발자와 사용자 간의 순시적인 정보교환 등에 응용 가능할 것이다.

후 기

이 논문은 부산대학교 자유과제 학술연구비(2년)에 의하여 연구되었으며, 지원에 깊은 감사를 드립니다.

참고문헌

- [1] 2006, Savitch, W., *Absolute JAVA*, Addison Wesley, University of California, San Diego.
- [2] 2004, 越塚越一, *數値流體力學*, 培風館, 東京.
- [3] 2002, 峯材 吉泰, *JAVAによる 流體・熱流動の數値シミュレーション*, 森北出版株式會社, 東京.
- [4] 2006, 峯材 吉泰, *JAVAで學ぶ シミュレーションの基礎*, 森北出版株式會社, 東京.
- [5] 2006, 박종천, 이병혁, 김정후, "자유표면 유동 시뮬레이션을 위한 고정확도 수치도식의 검토," *한국해양공학회지*, 제20권, 제4호, pp.31-36.
- [6] 2003, 박종천, 강대환, 전호환, "저항감소를 위한 물체후방의 형상설계에 관한 LES 해석," *한국해양공학회지*, 제17권, 제5호, pp.1-10.
- [7] 2007, 김경성, 박종천, "JAVA를 이용한 Web-기반 유동해석 프로그램의 개발," *한국전산유체공학회 2007년도 춘계 학술대회 논문집*, pp.112-115.
- [8] 2007, 김경성, 박종천, "2차원 관내 유동 해석을 위한 Web-기반 JAVA 프로그램 개발," *2007년도 한국해양과학기술협의회 공동학술대회 초록집*, pp.742-746.
- [9] 2007, 김경성, 박종천, "2차원 굽은 관내 유동 해석을 위한 웹-기반 수치 시뮬레이션 프로그램 개발," *대한조선학회 2007년도 추계학술대회 논문집*, pp.777-782.