

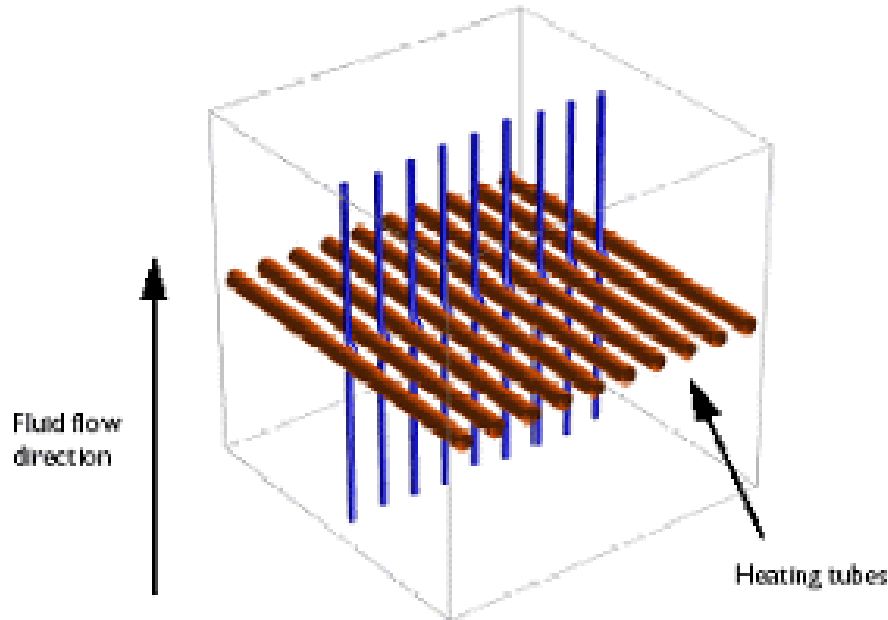
FEMLAB 사용법

FFMLAB 2.3은 MATLAB을 기본으로 하는 프로그램으로 MATLAB이 설치된 PC에서 설치가능하고 GUI(graphical user interface)방식을 사용하므로 사용자가 손쉽게 전산모사를 할 수 있다. 또 전산모사로 얻은 결과에 대해 사용자가 원하는 변수에 대한 시각화가 간편하다. 이러한 장점을 가진 FEMLAB를 간단한 예를 통해 사용법을 숙지하고자한다

1. 강제 및 자연 대류 열전달

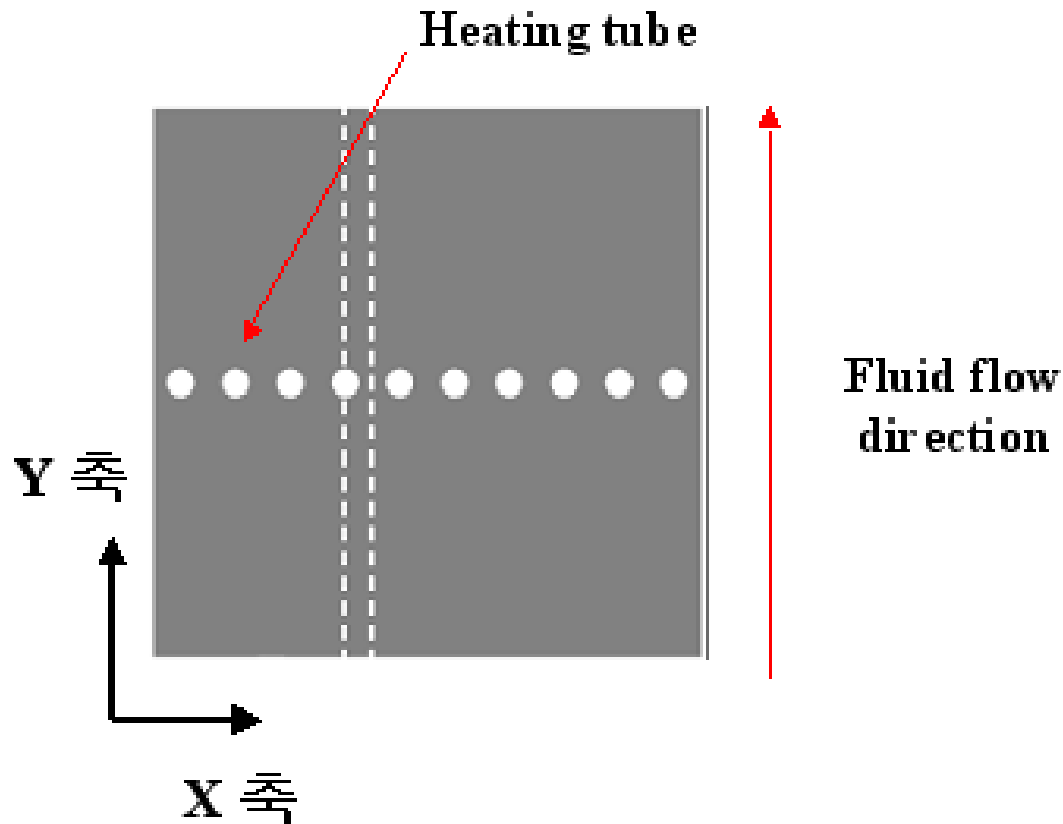
가. 모델 배경

유체장에서 열교환기를 가진 유체의 흐름 문제로서 밑에서 위로 20°C 의 유체(물)가 5×10^{-3} 로 흐르는 상압상태로 나가는 용기에 직경 0.5cm 인 여러개의 Heating tube(30°C)를 유체의 흐름방향의 수직방향으로 설치하였을 때 Heating tube 주위에서의 열의 분포와 유체의 거동을 알아보려고 한다. 아래 그림은 문제의 시스템을 시각화한 그림이다.



- 먼저 이 문제를 전산모사하기 전에 첫 번째로 고려할 점은 시스템의 차원(dimension)를 결정하는 것이다. 차원을 결정하기 위한 방법은 문제의 단순화 과정을 통하여 얻을 수 있다. 여기서는 용기 벽에서의 온도, 유체의 거동 영향을 무시한다고 가정하였다. 따라서 열교환기의 방향 즉 유체 흐름의 수직방향으로는 모두 같은 결과를 갖는다고 할 수 있으므로 유체가 흐르는 방향만 고려하면 되기 때문에 유체가 흐르는 방향과 수직방향만을 생각하여 2D라 결정하였다.

차원이 결정이 되면 대칭성을 찾아 geometry를 결정하게 된다. 이 문제의 경우 Heating tube를 중심으로 좌우 대칭성을 쉽게 찾을 수 있다. 그림에서 점선으로 표시한 부분을 geometry로 정해 전산모사를 한다.



나. 지배방정식

이 경우 두가지 물리적인 현상을 가진 모델로 열교환기주위에서의 온도 분포와 유체의 거동을 알기 위해서 편미분방정식(PDE, Partial Difference Equation)으로 *Incompressible Navier-Stokes* 식과 *Heat transfer* 식을 적용하였다.

두 가지 식은 4개의 미지의 변수를 갖는다. 속도장에서 속도의 x성분인 u , y성분인 v , 그리고 압력 p , 온도 T 가 있다. 이 4가지 변수는 구배(gradient)를 가지며 x성분과 y성분이 서로 상호관계를 가지고 있다.

먼저 단일성분의 *Incompressible Navier-Stokes* 식은 다음과 같다.

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \eta \nabla^2 \mathbf{u} + \mathbf{F}$$

$$\nabla \cdot \mathbf{u} = 0$$

여기서 \mathbf{F} 는 volume force, ρ 는 유체의 밀도(density), η 는 동점도(dynamic viscosity)이다.

다음으로 *Heat transfer* 식은 에너지 보존식으로 에너지의 변화는 heat source 와 heat flux의 차로 표현할 수 있다.

$$\rho c_p \frac{\partial T}{\partial t} + \nabla \cdot (-k \nabla T + \rho c_p T \mathbf{u}) = Q$$

여기서 c_p 는 유체의 열용량(heat capacity), ρ 는 유체의 밀도(density)이다. 괄호안은 heat flux vector로 확산과 대류항의 합으로 표현되고 대류항은 속도벡터(\mathbf{u})에 비례한다. 또, Q 는 heat source이다.

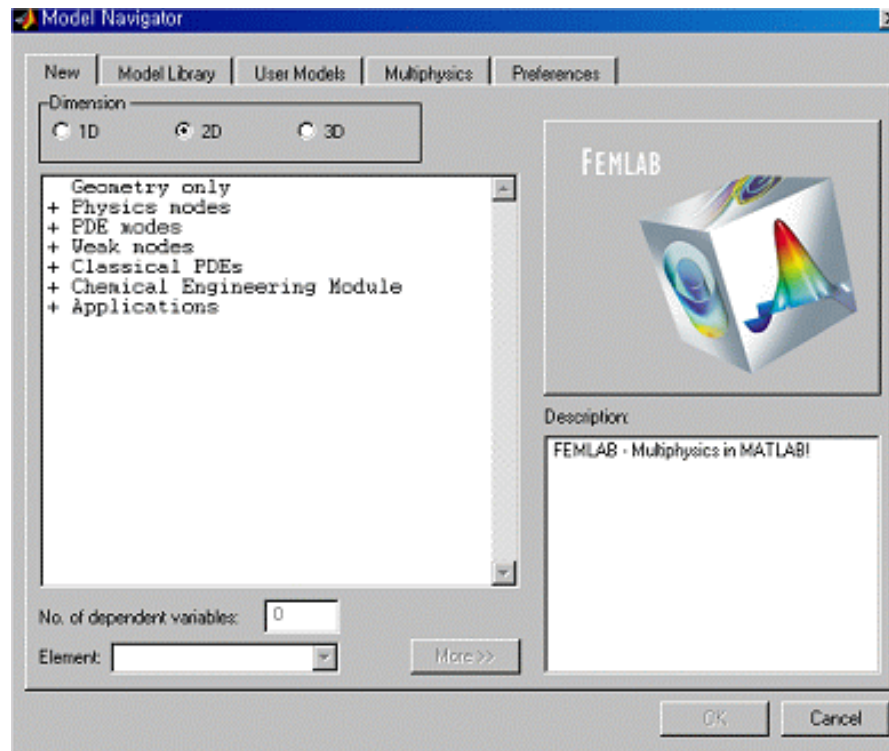
이 *Incompressible Navier-Stokes* 식과 *Heat transfer* 식은 속도벡터에 의해 상호연관되어있다.

다. FEMLAB에서의 모델 수립

1) FEMLAB에서 PDE 적용

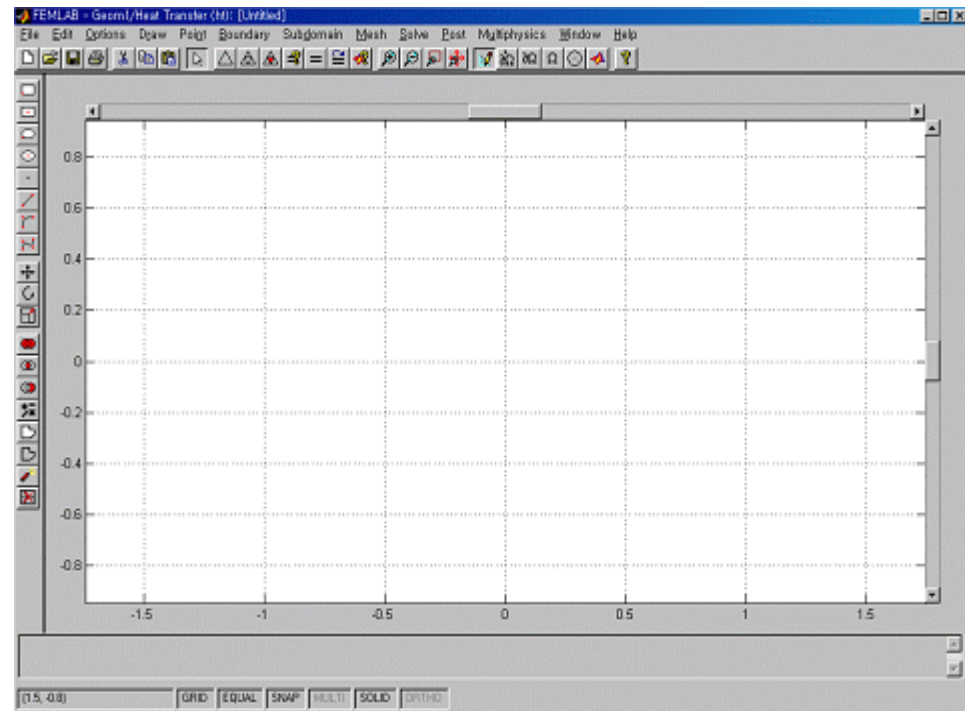
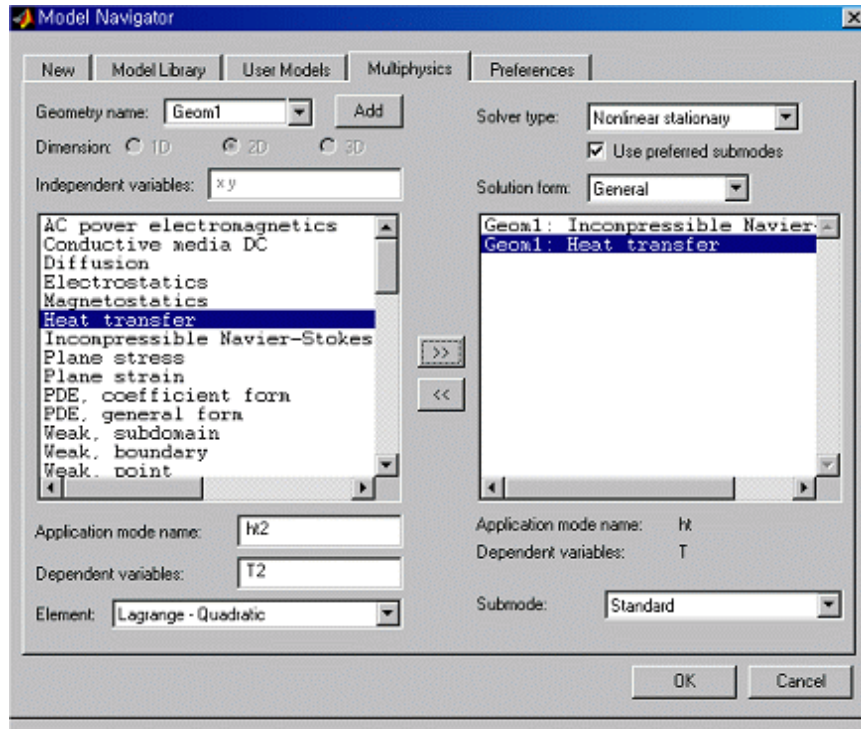
차원(dimension)과 geometry이 정해지고 적용하고자하는 PDE를 정하였으므로 이제 FEMLAB에 적용해보도록 하자.

- 우선 FEMLAB를 실행시키기 위해서는 바탕화면의 FEMLAB아이콘을 클릭하거나 MATLAB command 창에서 'femlab'이라 타이핑하면 실행이 된다.



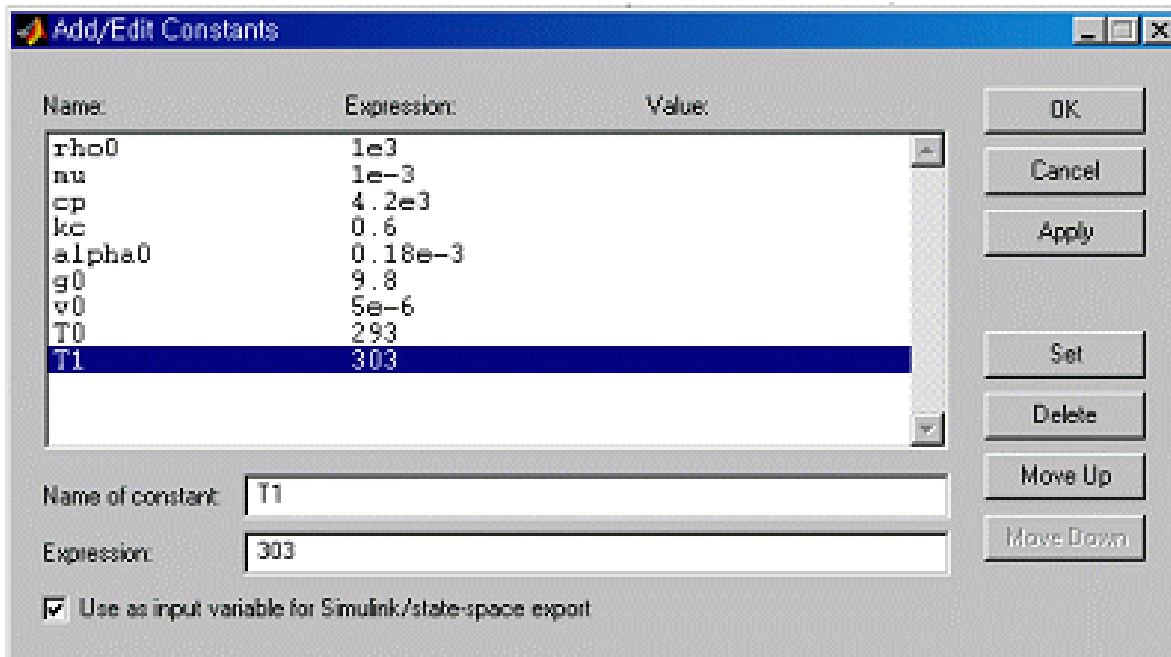
- 정의한 geometry의 차원이 2D이므로 버튼을 클릭한 후 *Multiphysics* 페이지로 이동한다. *Multiphysics* 페이지는 여러 PDE가 나열되어 사용자가 원하는 PDE를 적용하는 기능을 가지고 있다. 또 다성분계의 모델에서 PDE를 적용하였을 때 적용한 식과 변수들이 혼란의 소지가 있을 경우 사용자의 취향에 따라 *Application mode name*에서 식들의 이름을 정의할수 있고 *Dependent variable*에서 변수들의 이름을 정의할수 있다.
- 왼쪽의 PDE 리스트에서 적용하고자하는 *Incompressible Navier-Stokes* 선택한 후 왼쪽 화살표 버튼(>>)을 사용하여 적용한다. 기본값으로 *Application mode name*에 *ns*와 *Dependent variable*에 *u, v, p*라 정의 되어있다. 같은 방법으로 *Heat transfer*를 선택하여 추가하였고 식의 이름과 변수는 *ht*와 *T*라 정의 되어있다.

선택한 것들을 모두 확인한 후 *OK* 버튼을 누르면 Model Navigator가 닫히고 메인 화면이 생긴다.



2) 물리적 성질 입력과 격자(Grid) 생성

- PDE를 적용한 다음 모델의 전체적인 성질들을 적용하여야 한다. 이 모델에서는 유체의 물리적인 성질들과 입구부근에서의 온도, 열교환기 표면에서의 온도, 입구부근에서 유체의 속도들을 *Add/Edit Constants*에 입력해야한다. FEMLAB에서 쓰이는 모두 단위는 SI단위를 사용하고 있다.
- FEMLAB의 상단 메뉴바에서 *Option*를 선택한 후 하위 메뉴로 *Add/Edit Constants*를 선택하면 다음의 창이 열린다.



변수의 이름을 *Name of constant*에 기입하고 값을 *Expression*에 기입하면 된다. 변수값이 다른 변수의 함수일 때도 그 함수적 표현을 기입.

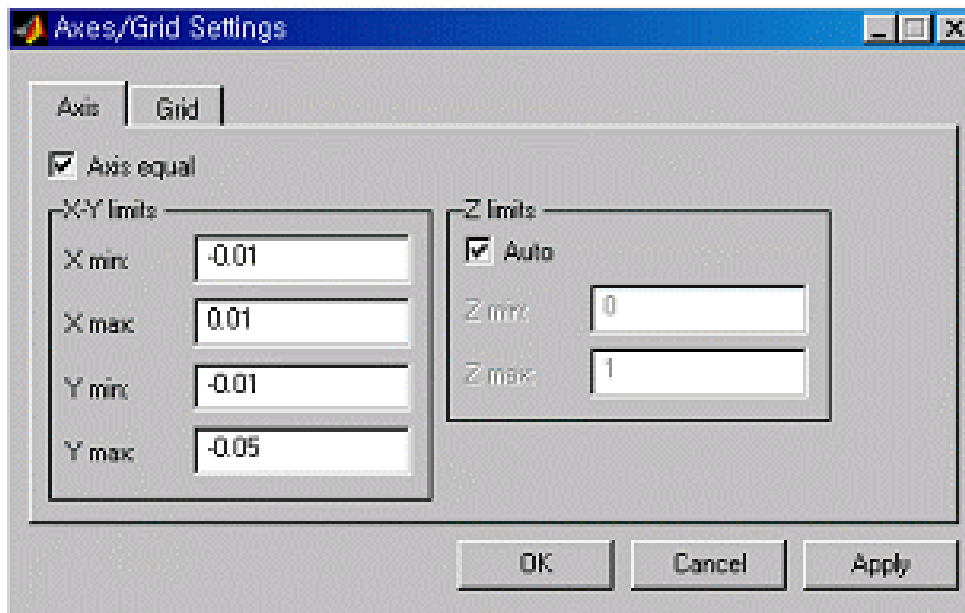
각각의 변수는 아래 표와 같이 기입하고 *OK* 버튼을 누른다.

Parameter	Symbol	Expression	Value
the density of the fluid	ρ	rho0	1×10^3
the dynamic viscosity	η	mu	1×10^{-3}
the heat capacity	c_p	cp	4.2×10^3
the thermal conductivity	k	kc	0.6
the volume expansion coefficient	α	alpha0	0.18×10^{-3}
the acceleration of gravity	g	g0	9.8
the inlet velocity	v_0	v0	5×10^{-3}
the temperature at the inlet	T_0	T0	293
the temperature on the heater	T_1	T1	303

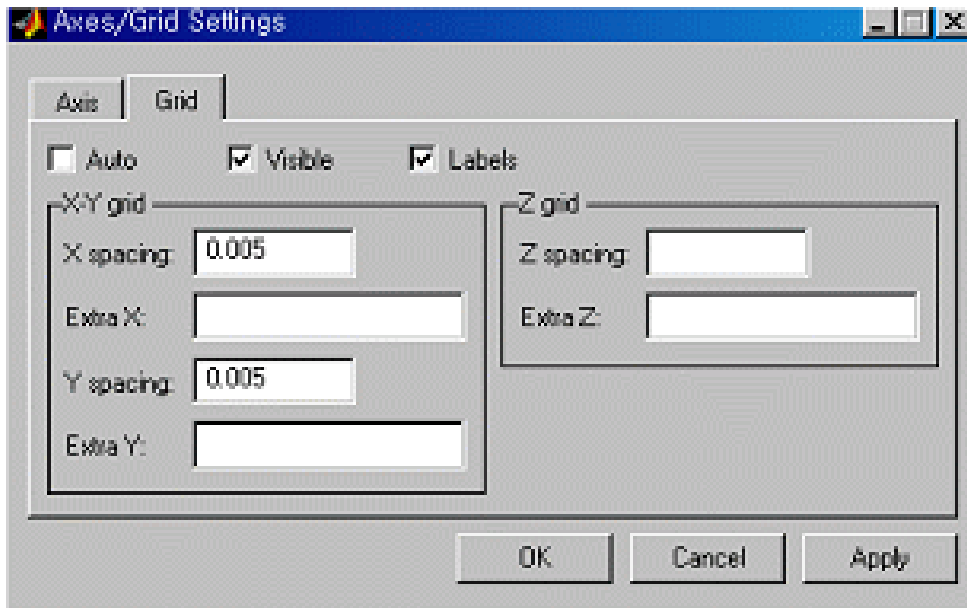
FEMLAB은 기본적으로 격자(*grid*)가 미터단위이므로 사용하기 쉽게 정의하고자하는 geometry에 맞도록 격자를 조정.

상단 메뉴바에서 *Option*을 선택한 후 하위 메뉴에서 *Axes/Grids Settings*를 선택한다.

창이 열리면 *Axis* 페이지로 이동하여 표현하고자 하는 geometry의 사이즈에 맞게 좌표를 정하게 된다. 이때 *Axis equal*를 비활성을 하면 비대칭적으로도 표현 가능하다.

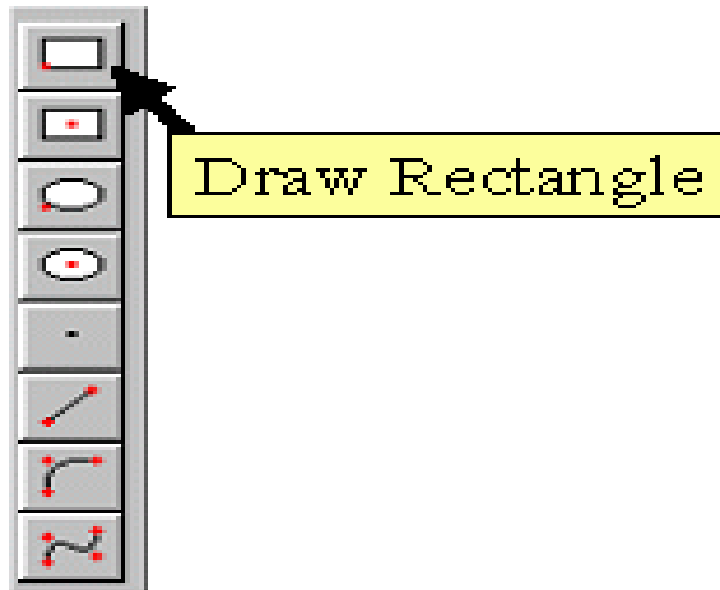


다음으로 *Grid* 페이지로 이동하여 *Auto*를 비활성화 시킨 후 사용자가 원하는 격자의 크기를 기입할 수 있다. 또 규칙적인 격자에 이외에 따로 추가하고 싶은 격자는 *Extra X*와 *Y*에 기입하여 표현할 수 있다. 여기서는 *X spacing*과 *Y spacing*에 각각 0.005를 기입하여 격자의 크기를 0.5 cm로 하였다.

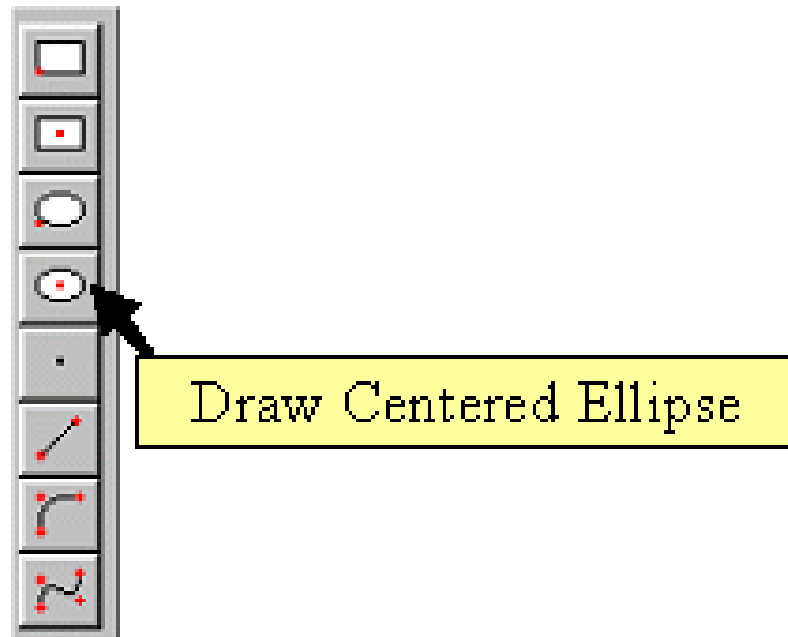


3) Geometry 생성

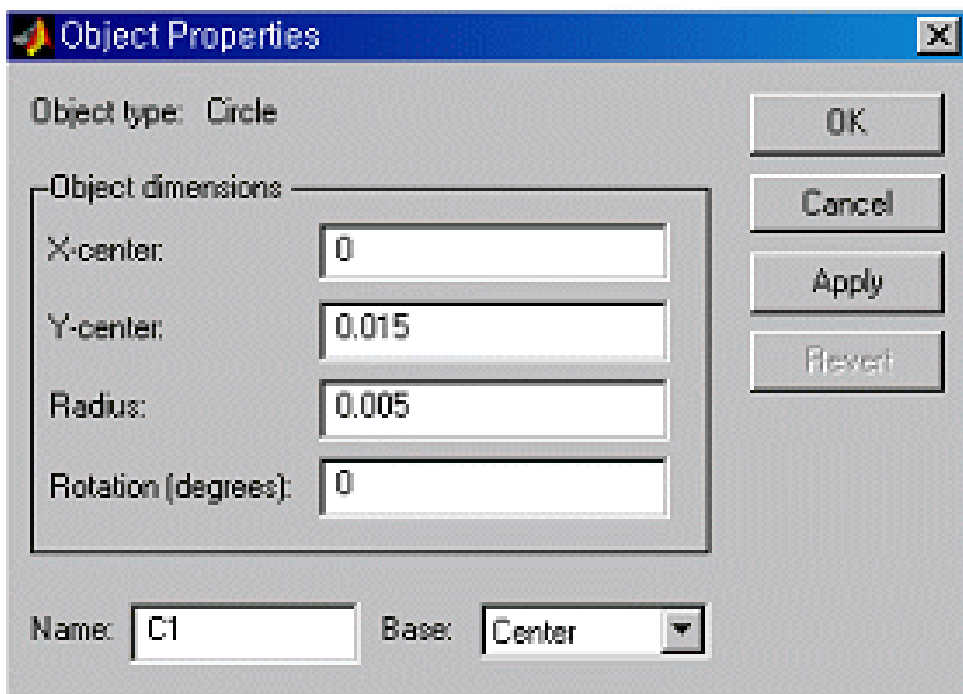
- 먼저 직사각형을 그려보자. 왼쪽의 그리기 도구모음을 보면 여러 가지 모양의 geometry를 생성할 수 있는 아이콘들이 배열되어있다. 여기서 *Draw Rectangle*를 선택한 후 마우스의 왼쪽 버튼을 사용하여 시작점을 (0,0)으로 하여 (0.005, 0.04)까지 마우스 버튼을 누른 상태로 drag 하면 직사각형이 생성된다.



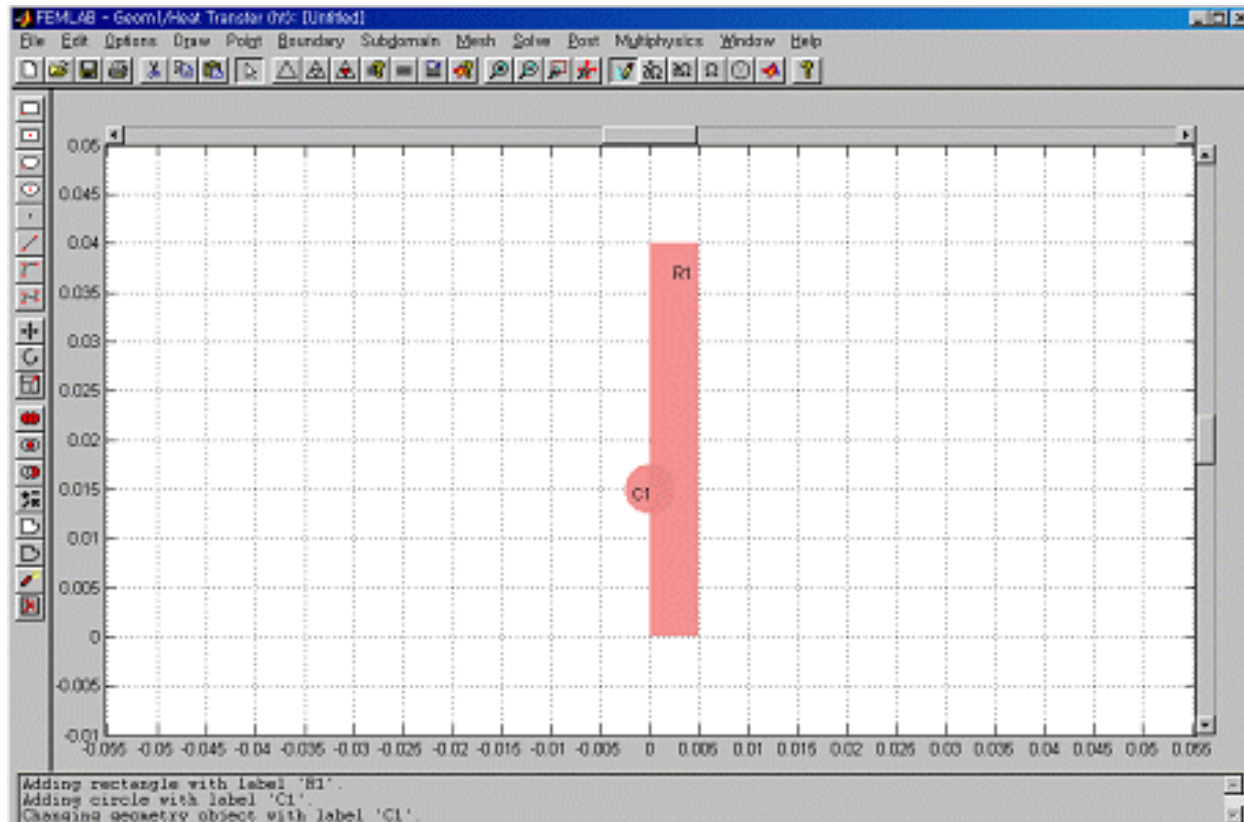
- 다음으로 직경이 0.005이고 중심이 (0, 0.015)인 원을 그리기 위해 옆의 도구모음에서 *Draw Centered Ellipse*를 선택한다. 마찬가지로 원점을 (0,0.015)로 시작하여 원을 그린다. 여기서 주의 할 점은 마우스 오른쪽 버튼을 눌러 반경을 그려야한다.



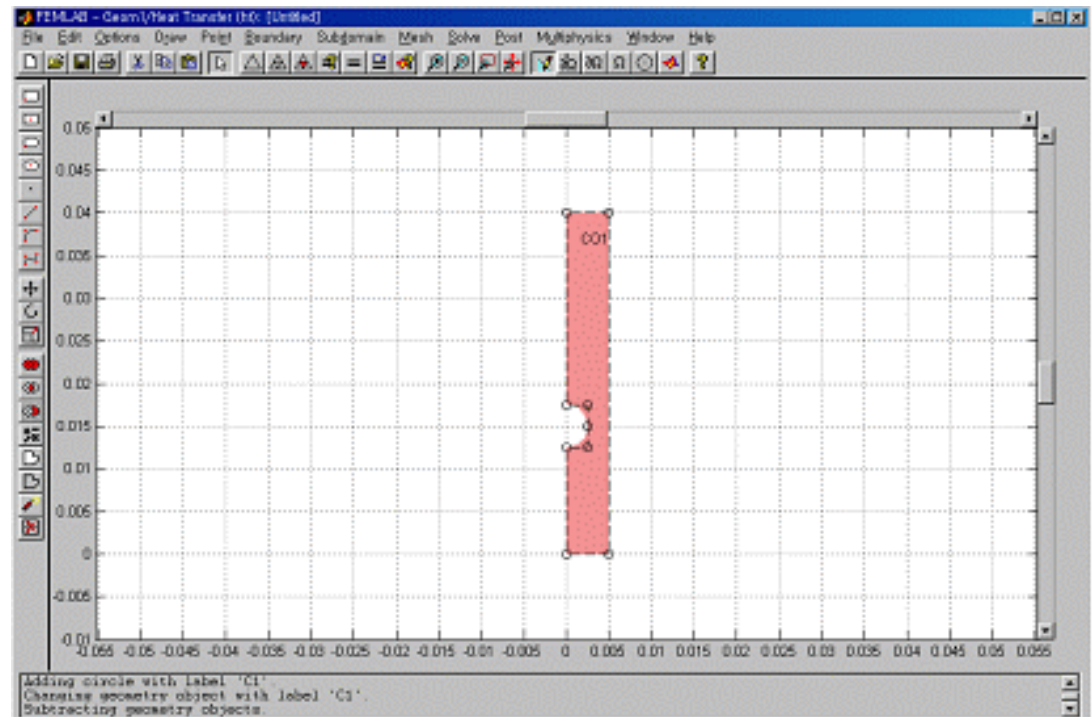
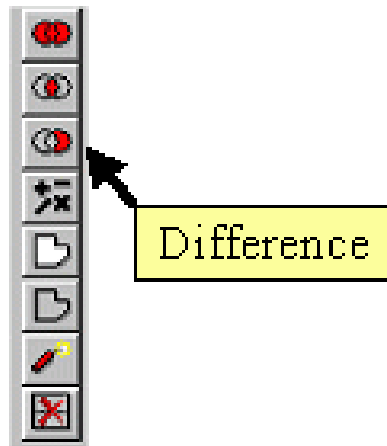
- 격자의 크기가 0.005이므로 한번에 반경이 0.0025인 원을 그릴 수 없다. 따라서 반경이 0.0025인 그리기 위해 원을 표현하기 위해 그린 원을 두 번 클릭하면 *Object Properties*창이 생긴다. 여기서 원하고자 하는 센터와 반경을 기입할 수 있다. 여기서 격자를 생성할 때 Extra X, 와 Y를 각각 (-0.0025, 0.0025), (0.0125, 0.175)를 생성하면 한번에 반경 0.0025인 원을 그릴 수 있다.



- 마우스로 drag 하여 직사각형과 원을 모두 선택하자. 이는 단축키 Ctrl-a를 이용하여 선택할 수도 있다.

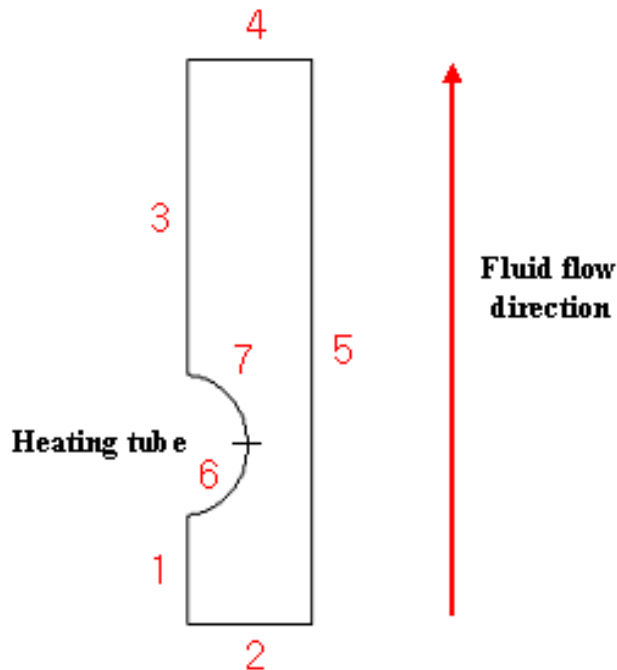


- 직사각형에 원이 함몰된 것을 표현하기 위해 옆의 그리기 도구모음에서 *Difference*를 선택한다.

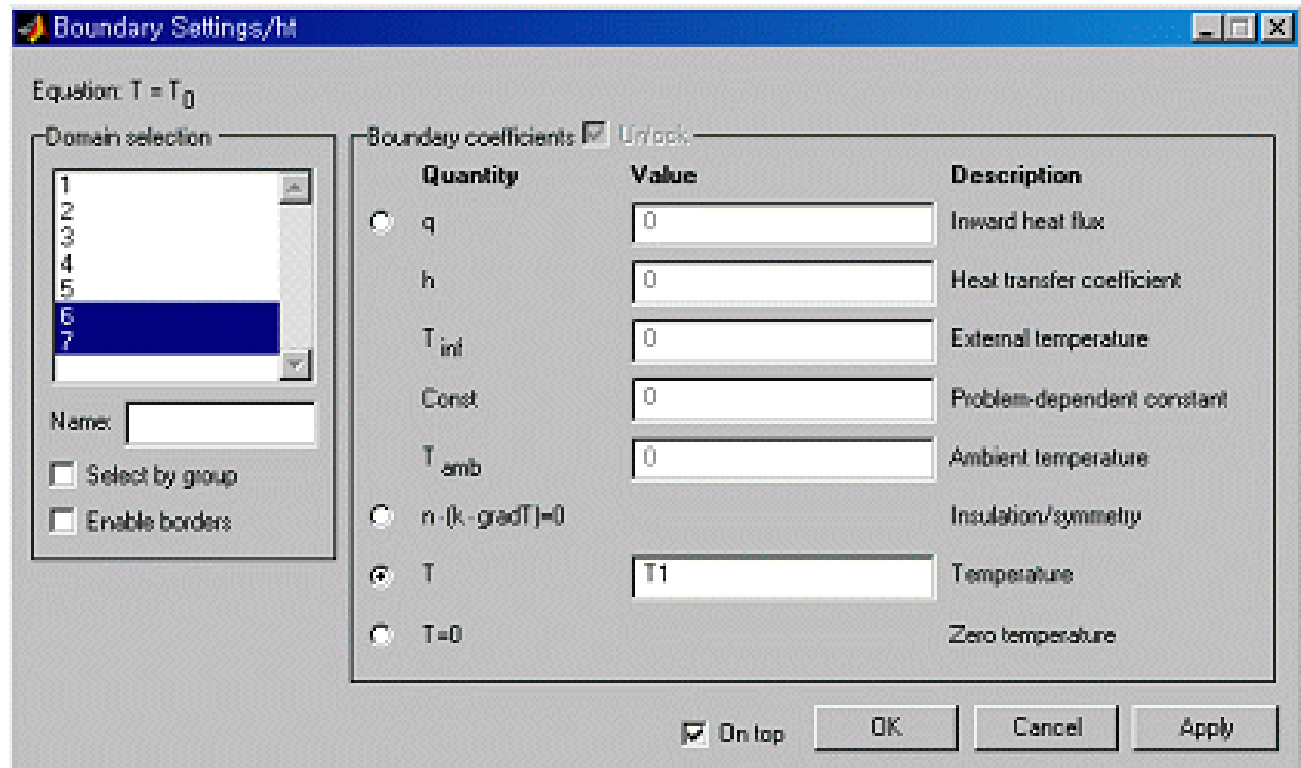


4) 경계조건 적용

- 지금 까지 우리가 풀고자하는 문제의 geometry를 생성하였다. 이제 이 geometry에 적용하고자하는 *Incompressible Navier-Stokes*식과 *Heat transfer*식에 대한 경계조건을 적용하고자한다.
- geometry의 경계는 총 7개이며 다음과 같이 정의되었다.

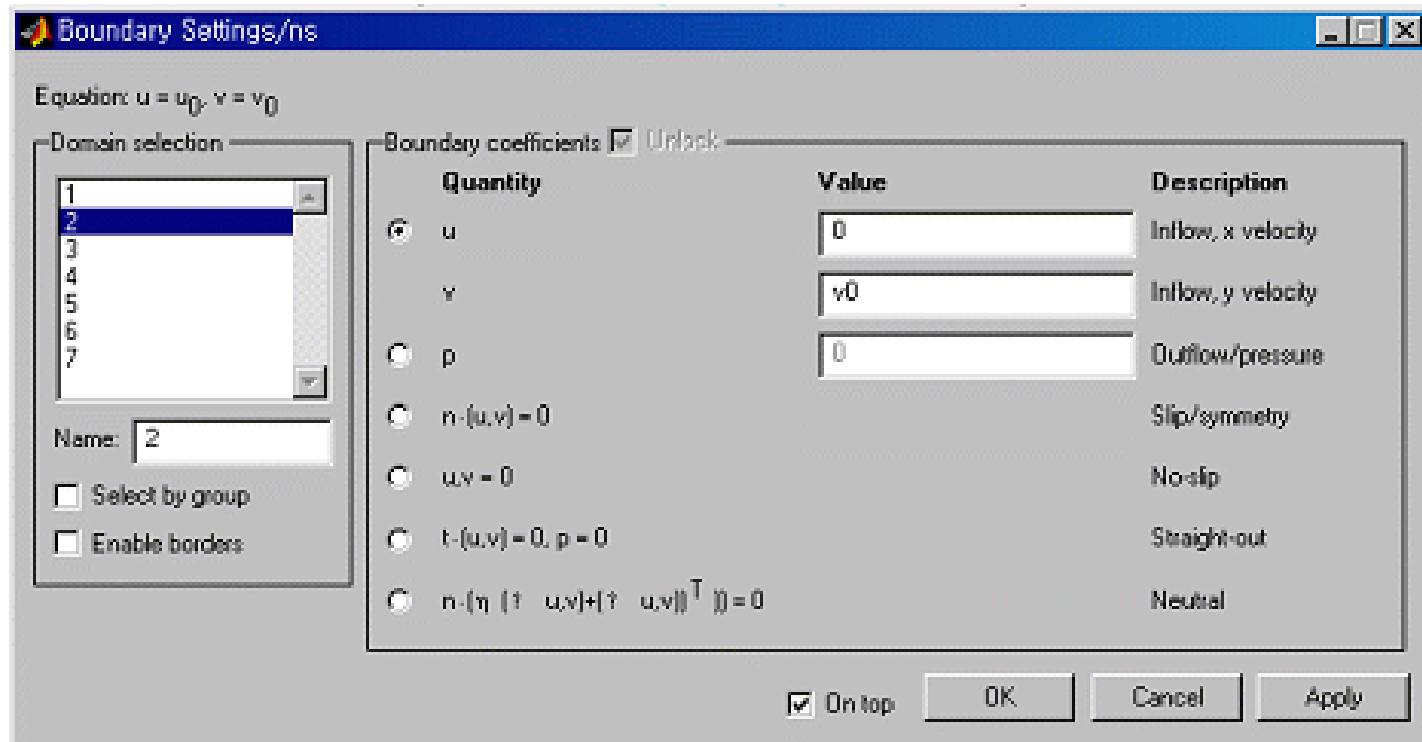


- 상단 메뉴바에서 *Multiphysics*를 선택하여 하위 메뉴에서 적용하고자 하는 PDE인 *Heat transfer*를 선택한다.
- 다음으로 상단 메뉴바에서 *Boundary*를 선택한 후 하위메뉴인 *Boundary setting*를 선택하면 *Boundary Setting*이라는 새로운 창이 뜬다. 왼쪽에 *Domain selection*에 경계들이 나열되어있고 여기서 경계를 선택할 수 있고 또는 화면상에서 원하는 경계를 클릭하여 선택할 수도 있다. 선택이 된 경계는 붉게 변하게 된다.



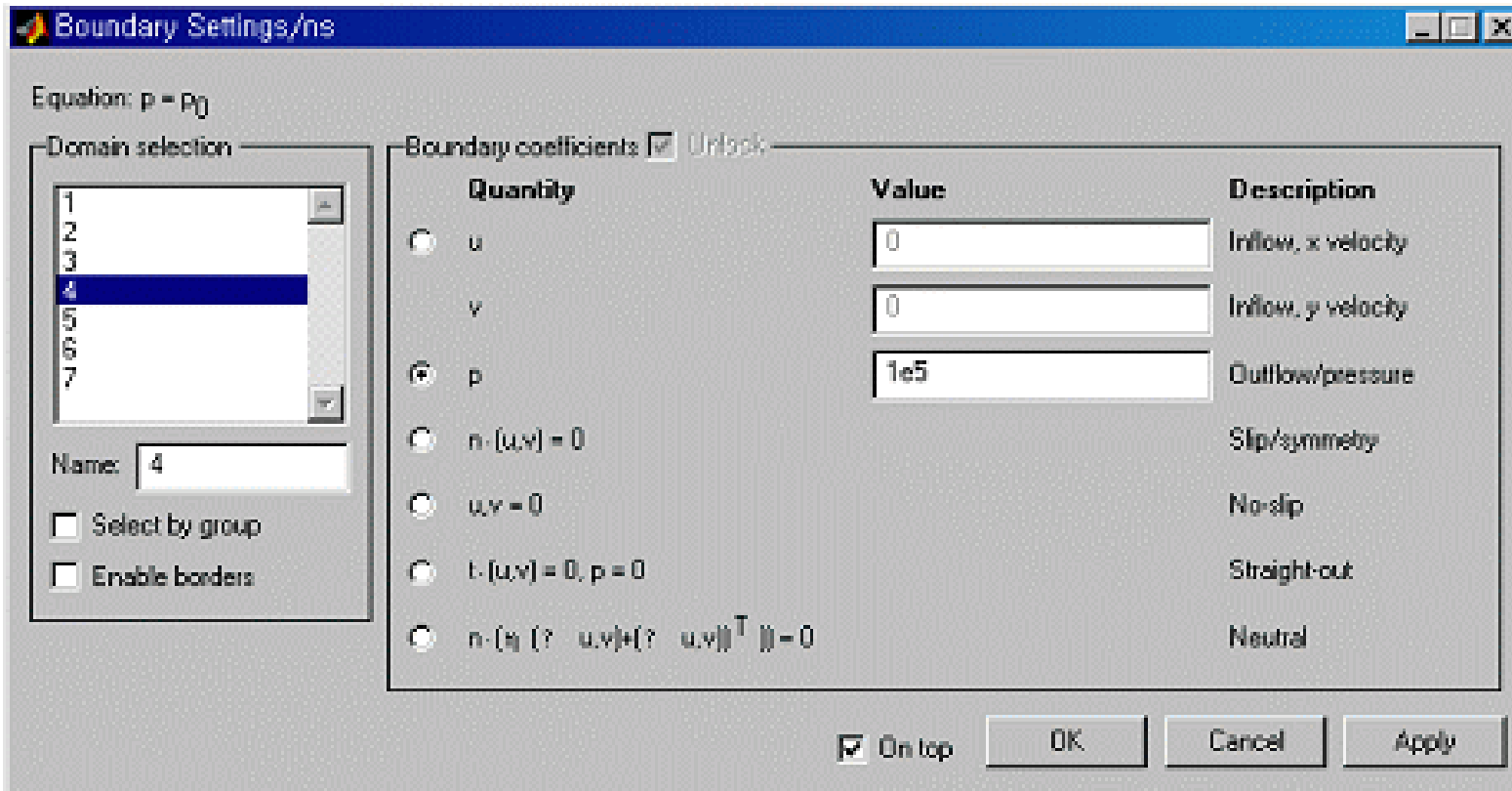
- 경계 2번은 유체가 주입되는 부분이므로 T_0 를 입력하였다.
- 경계 6, 7번은 *Heating tube*이므로 둘다 경계조건으로 T_1 를 입력하였다. ctrl키를 사용하여 6,7번 한번에 선택할 수 있다.
- 나머지의 경계는 *insulation/symmetry*를 적용하고 OK버튼을 누르면 된다.

- 다음으로 *Incompressible Navier–Stokes*식에 대한 경계조건들을 적용해보자.
- 메뉴에서 *Multiphysics*를 선택하여 하위 메뉴에서 적용하고자하는 *Incompressible Navier–Stokes*를 선택한다.
- *Heat transfer*의 경우와 동일한 방법으로 *Boundary setting*창을 띄운다.
- 경계 2번은 유체가 주입되는 부분으로 y방향으로 v_0 의 속도로 유체가 주입된다. 따라서 *x velocity*와 *y velocity* 칸에 0과 v_0 를 입력한다.



- Heating tube인 경계 6, 7번은 *No-slip*를 선택한다.
경계 4번은 유체가 나가는 출구로 상압조건인 1×10^5 을 입력한다.
나머지 경계에 대해서는 *Slip/symmetry*로 적용하자.

- 이상으로 두 PDE에 대한 경계조건을 모두 적용하였다.



5) subdomain의 정의

- 적용한 두 지배방정식의 계수들은 서로 상호관계를 가지고 있으므로 subdomain 모드에서 이를 적용해보자.
- 먼저 *Incompressible Navier-Stokes*식에 대해 생각해 보자.
- 상단 메뉴바에서 *Subdomain*를 선택하고 하위 메뉴로 *Subdomain setting*를 클릭하면 아래의 창이 열린다. 왼쪽에는 subdomain의 번호와 오른쪽에는 PDE의 계수들이 나열되어있다. 이번의 경우는 subdomain이 하나이므로 1만 나열되어있다. 여기서도 메인화면에서 사용자가 직접 마우스로 선택할 수도 있다.
-

Subdomain Settings/ns

Equation: $-\nabla \cdot \eta(\nabla u + (\nabla u)^T) + \rho(u \cdot \nabla)u + \nabla p = F; \nabla \cdot u = 0$

Coefficients | Init | Element

Domain selection

1

Name: 1

Select by group

Active in this domain

PDE coefficients Unlock

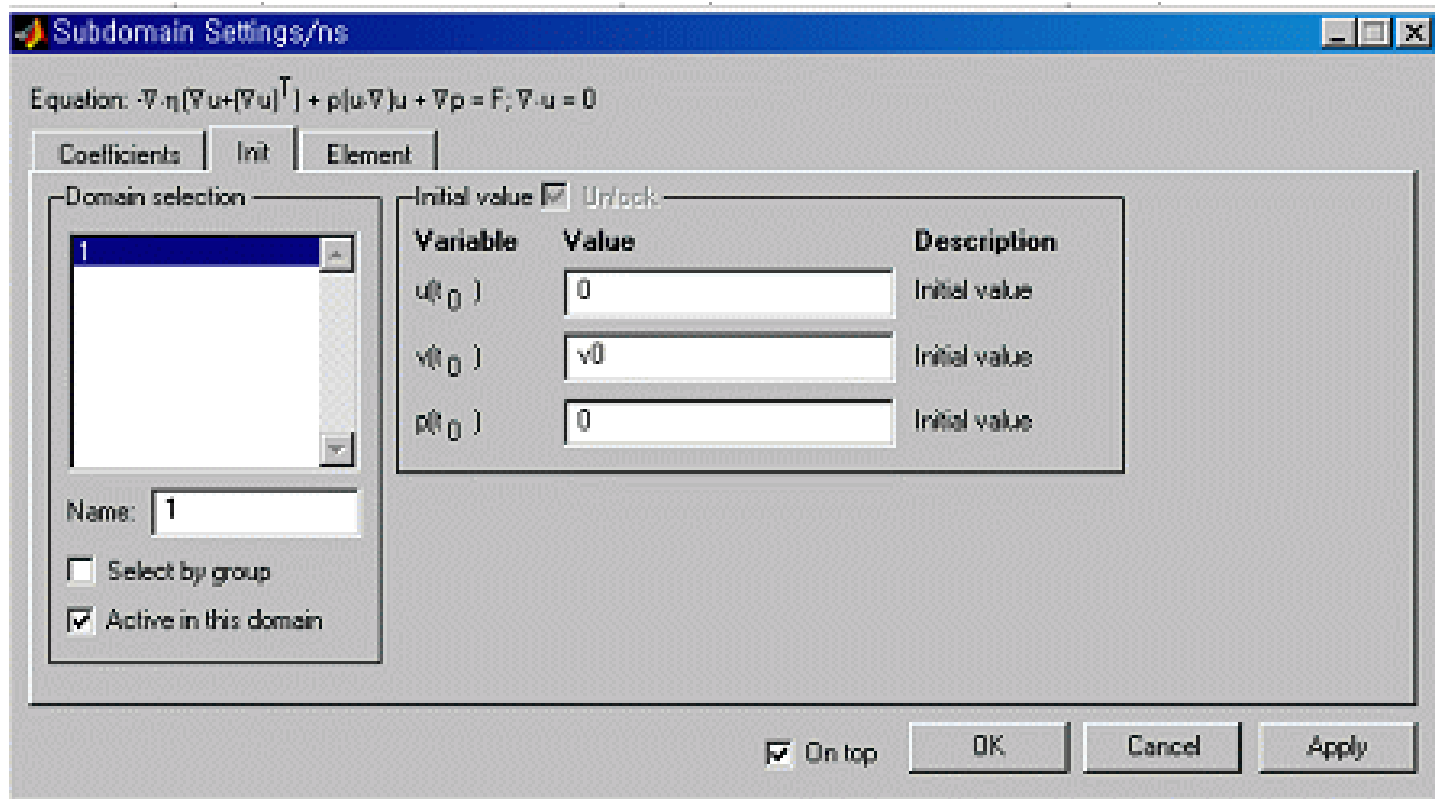
Coefficient	Value	Description
ρ	rho0	Density
η	mu	Dynamic viscosity
F_x	0	Volume force, x-dir.
F_y	alpha0*g0*rho0*(T-T0)	Volume force, y-dir.

Streamline diffusion

Streamline Diffusion Settings...

On top OK Cancel Apply

- 유체가 주입되는 방향은 y축 방향만이고 온도(T)는 *Heat transfer* 식과 서로 상호관계에 있는 것을 상기하자.
- 초기값을 *Subdomain Settings*에서 init 페이지로 이동하여 초기값을 기입한다. 옳은 초기값 기입은 해의 수렴을 일으켜 해를 구할 수 있다. 여기서는 y 방향의 속도성분을 v_0 라 하였다.



- 다음으로 *Heat transfer*의 경우에 대해 *Subdomain*를 고려해보자.
- *Incompressible Navier–Stokes*경우와 마찬가지로 *Subdomain setting* 창을 열고 아래와 같이 입력한다. T_x 와 T_y 는 각 방향의 온도구배를 나타내는 것이다.

Coefficient	Value	Description
ρ	ρ_0	Density
c	c_p	Heat capacity
k	k_c	Thermal conductivity
Q	$-c_p \cdot \rho_0 \cdot (T_x \cdot u + T_y \cdot v)$	Heat Source

Equation: $-\nabla \cdot (k \nabla T) = Q + h_{\text{trans}}(T_{\text{ext}} - T) + C_{\text{trans}}(T_{\text{ambtrans}}^4 - T^4)$, T = temperature

Coefficients | **Init** | Element

Domain selection

1

Name:

Select by group

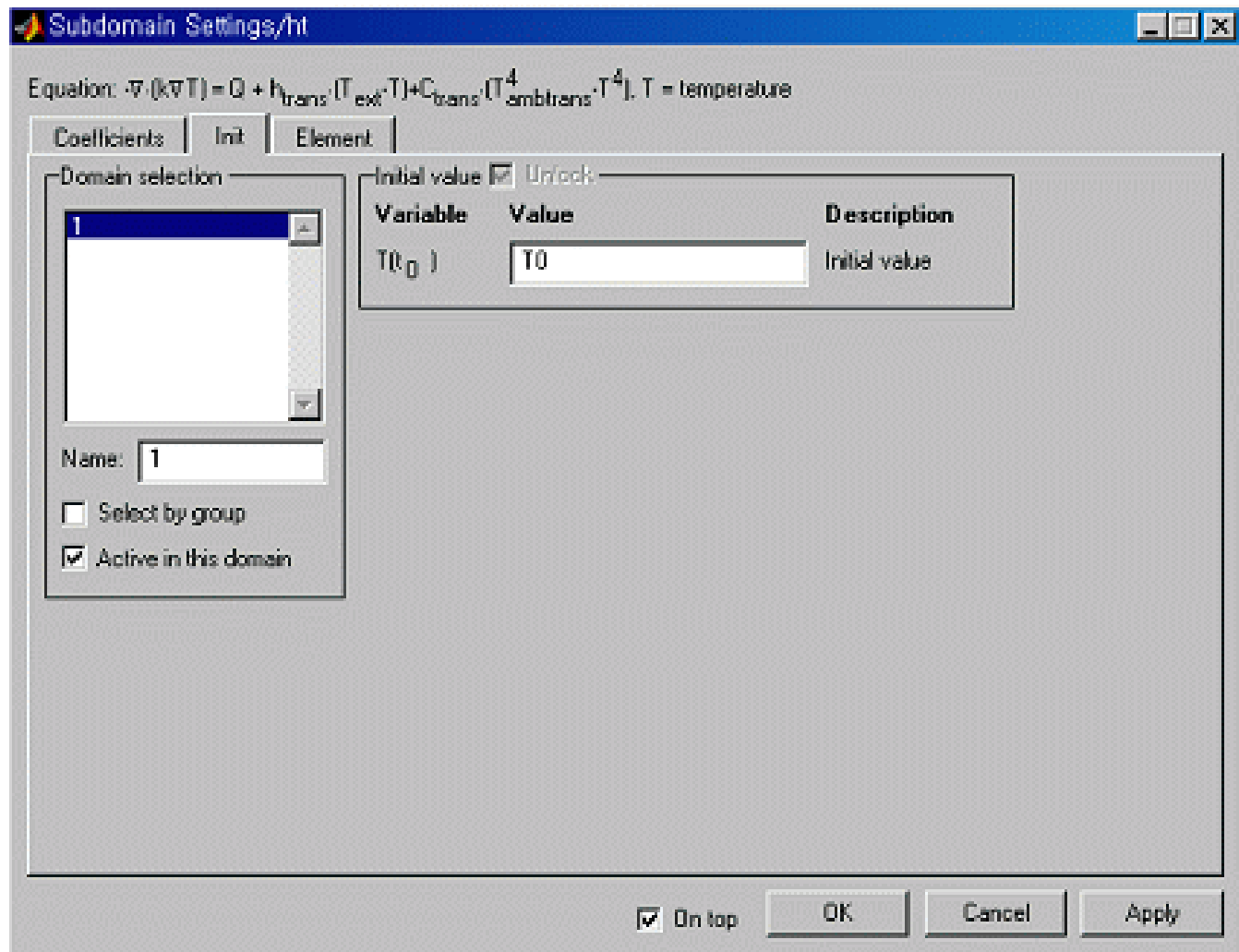
Active in this domain

PDE coefficients **Unlock:**

Use defined material:

Coefficient	Value	Description
ρ	<input type="text" value="rho0"/>	Density
C	<input type="text" value="cp"/>	Heat capacity
k	<input type="text" value="kc"/>	Thermal conductivity
Q	<input type="text" value="-cp*rho0*(Tx'u+Ty*v)"/>	Heat source
h_{trans}	<input type="text" value="0"/>	Convect. heat transf. coeff.
T_{ext}	<input type="text" value="0"/>	External temperature
C_{trans}	<input type="text" value="0"/>	User-defined constant
T_{ambtrans}	<input type="text" value="0"/>	Ambient temperature

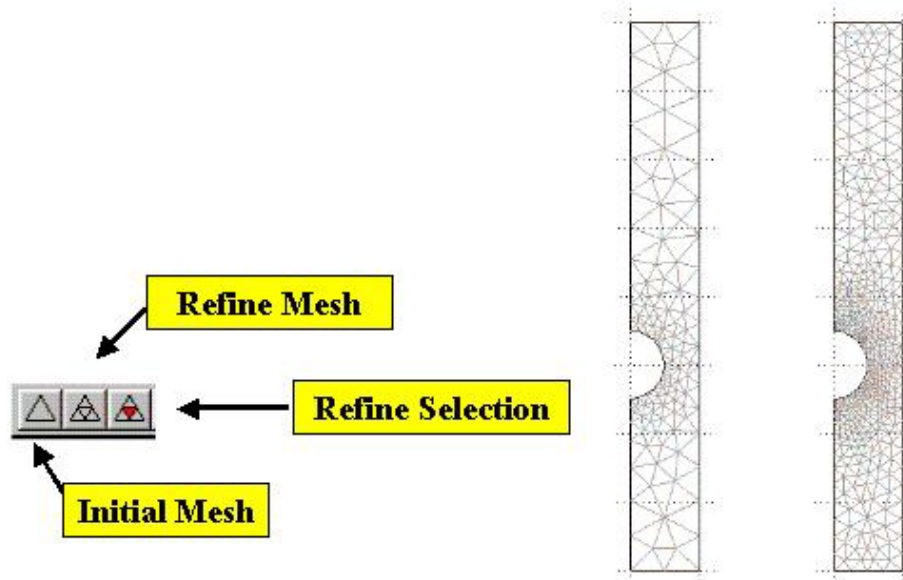
*Heat transfer*와 동일하게 초기값을 입력할 수 있다. 주입되는 유체의 온도는 T_0 라 입력하였다



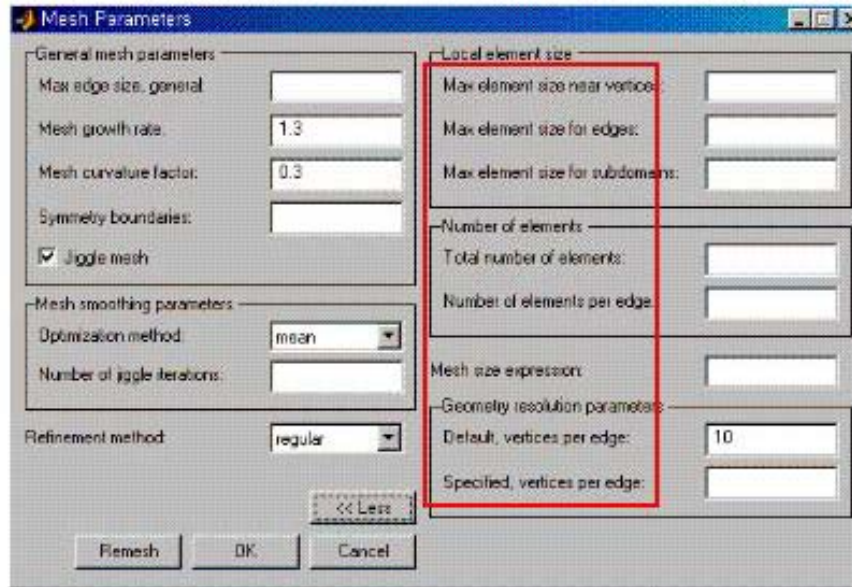
6) Mesh 생성

FEMLAB은 유한 요소법(finite element method)에 기초하여 해를 구하기 때문에 geometry를 작은 mesh로 나뉘어야한다. *Mesh* 버튼을 눌러 기본적인 mesh를 생성시킬수 있다. 그리고 *Solve* 버튼을 누르면 작동적으로 기본적인 mesh를 생성하여 해를 구한다.

상단의 도구 모음에서 *Initial Mesh* 버튼 또는 *Mesh Mode* 버튼을 클릭하게 되면 기본적인 mesh가 생성된다. 여기서 더 작은 mesh를 생성하기 위해서는 손쉬운 방법으로 *Refine Mesh* 버튼을 누르면 더 작은 mesh가 생성된다. 아래그림에 왼쪽은 기본적인 mesh를 생성한 것이고 왼쪽은 더 작은 mesh를 생성한 그림이다.



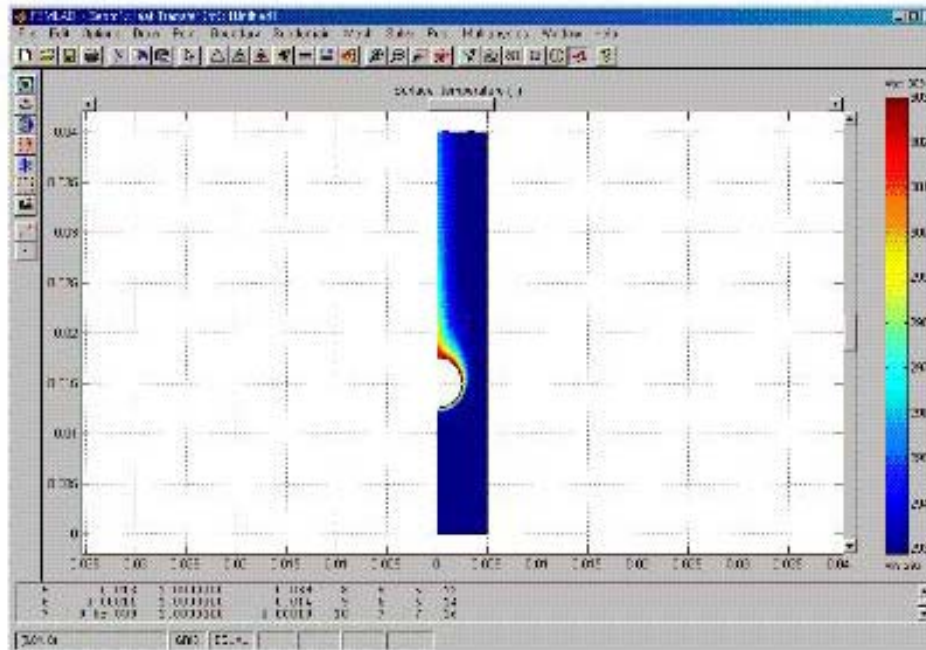
- 만약 사용자가 원하는 부분에 대해 다른 mesh상태를 원하면 메뉴바에서 *Mesh Parameter*를 선택하여 원하는 경계를 입력하고 mesh의 크기를 입력하여 mesh를 생성할 수 있다.
- 그리고 메뉴에서 *Solver Parameters*를 선택하여 사용자가 원하는 형태의 해를 얻을 수 있다. 그 예로 time-dependent를 선택하고 출력시간을 입력하면 입력한 시간동안 정의한 시스템안에서의 해의 거동을 볼 수 있다. 이는 다른 동영상 파일로 출력 가능하다.



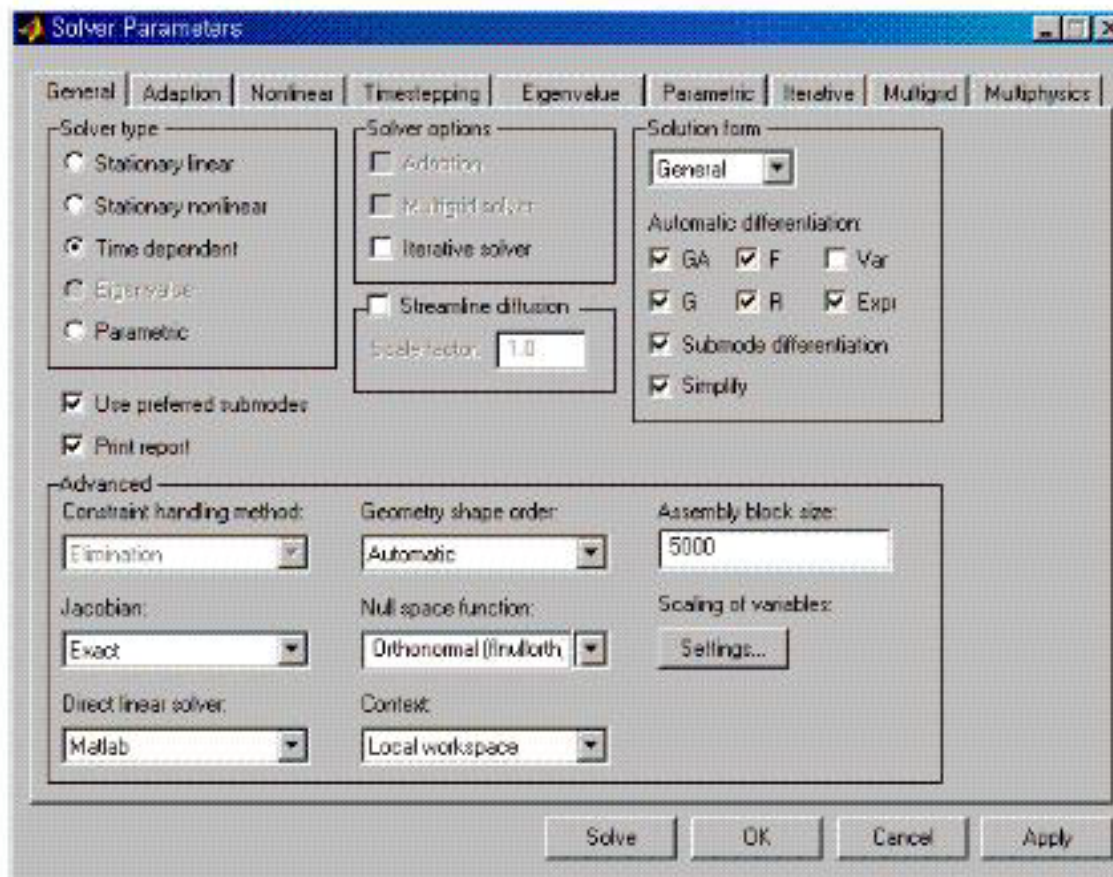
7) 해 구하기

위 과정들을 모두 수행한 사용자들은 상단의 도구모음에서 *Solve Problem* 버튼을 클릭함으로써 손쉽게 해를 구할 수 있다.

운영체제 Windows 2000에서 CPU 1.5-GHz Pentium 4를 가진 컴퓨터에서 Solve Problem 버튼을 누르면 이 문제의 경우 약 19초후 해를 얻는다.



그리고 메뉴에서 **Solver Parameters**를 선택하여 사용자가 원하는 형태의 해를 얻을 수 있다. 그 예로 time-dependent를 선택하고 출력시간을 입력하면 입력한 시간 동안 정의한 시스템안에서의 해의 거동을 볼 수 있다. 이는 다른 동영상 파일로 출력 가능하다.



8) 시각화 모드 (*Post mode*)

- *Post* 모드에서 추가적으로 plot 타입을 추가할 수 있고 다른 형태의 plot를 얻을 수 있다. 원하는 변수의 해와 다양한 좌표의 시각화는 다른 소프트웨어와 비교하였을 때 유용하다. 많이 쓰이는 표현은 postprocessing variables로서 정의 되어있다.
- 상단 도구 모음에서 *Plot Parameters* 버튼을 누르면 메인화면에 *Plot Parameters* 창이 열린다.
- *General* 페이지에서 Plot type를 결정할 수있다. 두 가지 이상의 변수에 대하여 표시할 때는 Plot type에서 두가지 이상를 체크하고 각각의 페이지에서 그 변수를 정의하면 된다. 이 경우 *Surface* 와 *Arrow* 두 type를 체크하였고 Arrow 페이지에서 *x velocity(u)*와 *y velocity(v)*를 선택하였다. 그리고 *X spacing* 과 *Y spacing* 에 각각 10과 15를 입력하였다. 화살표의 스타일과 색도 선택할 수 있다.



Plot Parameters

Plot Parameters

General | Surface | Contour | Line | Arrow | Deform | Flow | Animate

Plot type:

- Surface
- Contour
- Line
- Arrow
- Deformed shape
- Flow lines
- Geometry boundaries

Solution at time: none

Solution at angle (phase): 0 degrees

Renderer: ZBuffer

Smooth Settings...

Element refinement: Auto 3

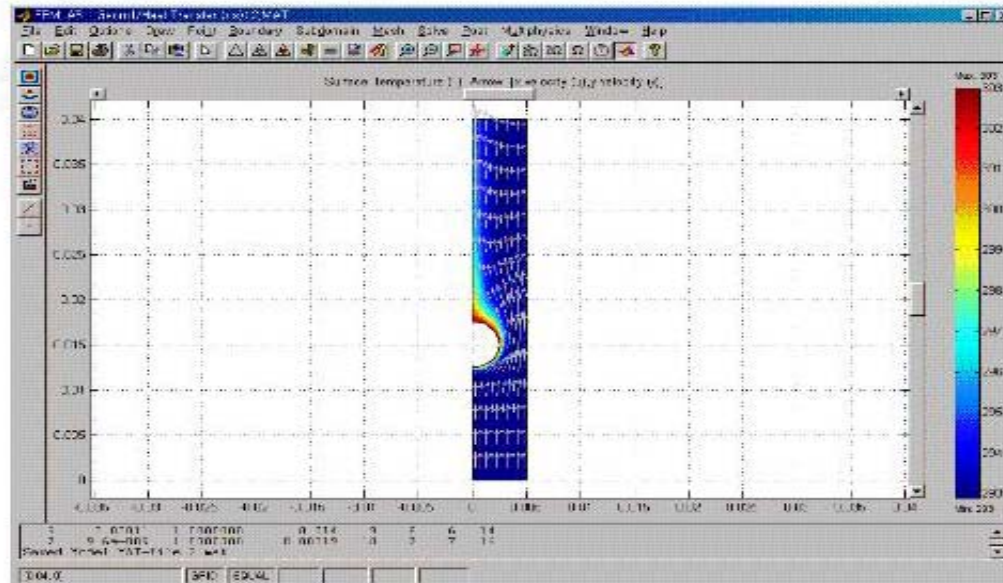
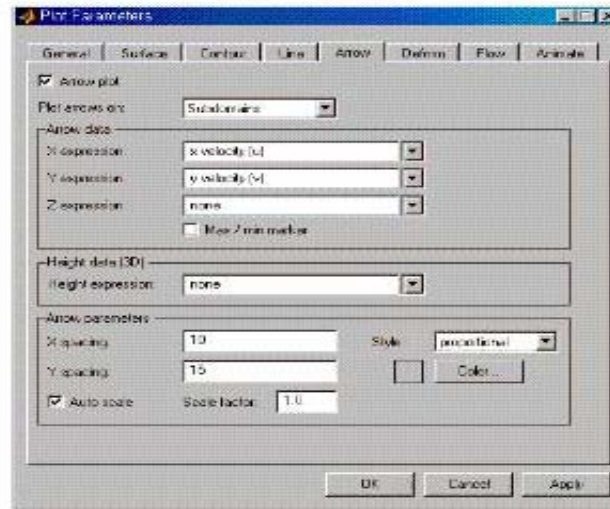
Title:

Auto Surface: temperature (T) Arrow: (heat flux (flux_x_h)) heat flux (flux_y_h)

Plot in main GUI axes

Plot solution automatically

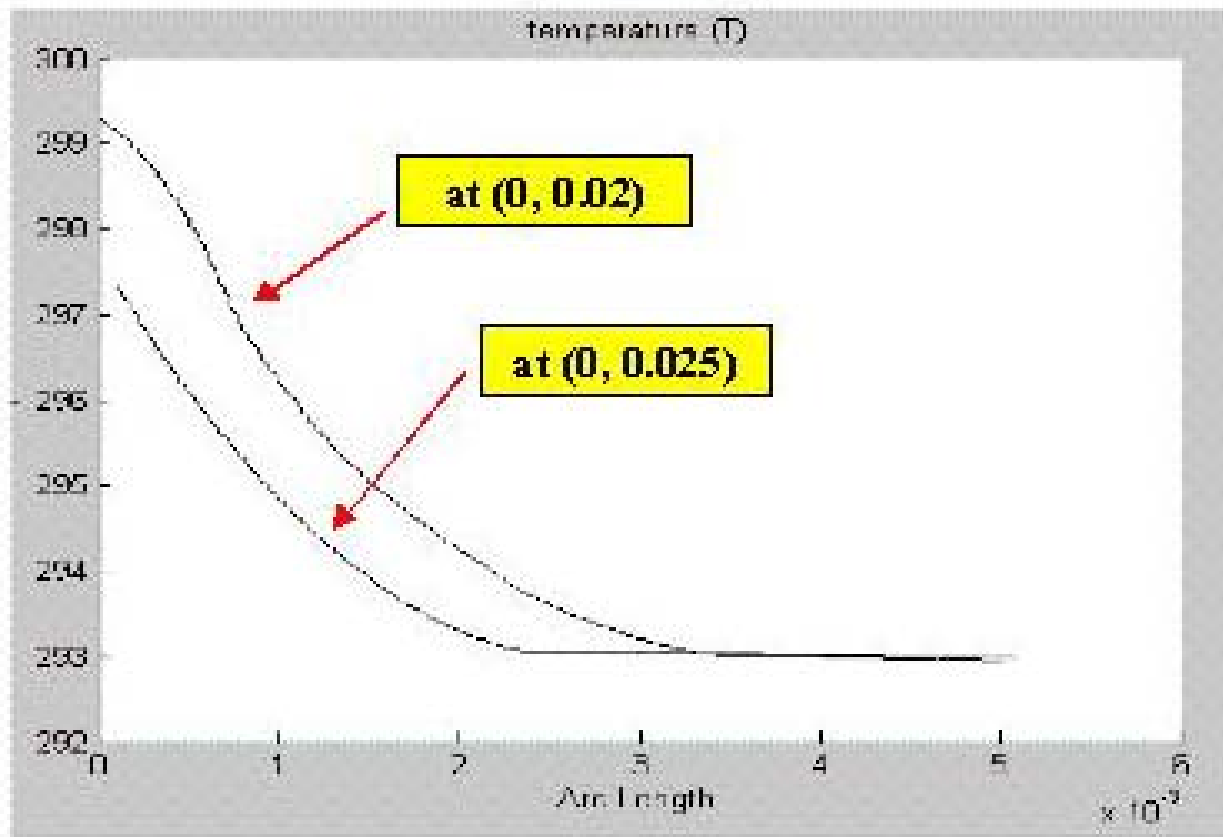
OK Cancel Apply



위 그림에서 surface는 온도의 분포를 표시하는 동시에 화살표로 유체의 거동을 시각화 한 것이다.

9) Advanced Postprocessing

- FEMLAB은 geometry의 단면에 대해서도 쉽게 해를 도식화할 수 있다. 이는 옆의 도구 모음에서 *Draw Line for Cross-Section Line Plot* 버튼을 눌러 수행할 수 있다. 그 예로 (0, 0.02)와 (0, 0.025)에서의 단면에서 온도 profile를 구해보자.
- 옆의 도구모음에서 *Draw Line for Cross-Section Line* 버튼을 누른 후 (0, 0.02)에서 수평으로 선을 긋으면 figure 창이 열린다. MATLAB 명령창에서 'hold on'이라 타이밍한 후 (0, 0.025)에서 수평을 긋자. FEMLAB은 MATLAB의 명령어를 기본적으로 적용된다.
- 이러한 작업은 메뉴에서 *Cross-section parameter*메뉴에서도 가능하다.
- *General* 페이지에서 *Line*를 체크하고 시작점의 좌표와 끝점의 좌표를 입력하고 *Apply* 버튼을 누르면 *Figure* 창이 열린다.



10) 출구부분에서의 평균온도 구하기

- 평균 출구온도를 알기 위해서는 시스템 출구부분에서 유체의 온도와 속도는 지점마다 다르므로 산술평균 개념을 도입하면 된다. 분자를 얻기 위해 *Post* 메뉴의 하위 메뉴로 *Boundary intergration*를 클릭 후 출구인 경계 4번을 선택하고 *Expression* 칸에 $T*v$ 라 입력하자.
- *Export result to workpace* 에 체크한 후 *Apply*를 누르면 MATLAB 작업창에 ans로 입력되어진다. 따라서 MATLAB prompt에서 $l_{tv} = ans$ 라고 타이핑하면 그 값이 l_{tv} 라는 방에 입력이 된다.
- 분모를 구하기 위해 같은 창에서 *Exprssion* 칸에 v 라 입력하고 *Apply*를 누르면 계산되어 진다. 다음과 같이 MATLAB 상에서 쓰면 $lv = ans$ $T_{mean} = l_{tv}/lv$
- 출구의 평균온도 293.8 K 라는 값을 얻을 수 있다. 공급된 유체의 온도보다 0.8 K 상승한 것을 알 수 있다.

FEMLAB Model M-file

```
% FEMLAB Model M-file
```

```
fclose fem
```

```
% FEMLAB Version
```

```
clear vrsn;
```

```
vrsn.name='FEMLAB 2.3';
```

```
vrsn.major=0;
```

```
vrsn.build=153;
```

```
fem.version=vrsn;
```

```
% Recorded command sequence
```

```
% New geometry 1
```

```
fem.sdim={'x','y'};
```

```
% Geometry
```

```
clear s c p
```

```
p=[0 0 0 0 0.0025000000000000001 0.0025000000000000001 ...
```

```
0.0025000000000000001 0.0050000000000000001 0.0050000000000000001;0 ...
```

```
0.012499999999999995 0.017499999999999995 0.040000000000000001 ...
```

```
0.012499999999999995 0.014999999999999996 0.017499999999999995 0 ...
```

```
0.040000000000000001];
```



```
rb={[1 2 3 4 6 8 9],[1 1 3 4 8;2 8 4 9 9],[2 3;5 7;6 6],zeros(4,0)};
wt={zeros(1,0),ones(2,5),[1 1;0.70710678118654746 0.70710678118654746;1 1], ...
zeros(4,0)};
lr={[NaN NaN NaN NaN NaN NaN NaN],[0 1 0 0 1;1 0 1 1 0],[0 1;1 0],zeros(2, ...
0)};
CO1=solid2(p,rb,wt,lr);
objs={CO1};
names={'CO1'};
s.objs=objs;
s.name=names;
objs={};
names={};
c.objs=objs;
c.name=names;
objs={};
names={};
p.objs=objs;
p.name=names;
drawstruct=struct('s','s','c','c','p',p);
fem.draw=drawstruct;
fem.geom=geomcsg(fem);
clear appl
```

```

% Application mode 1
appl{1}.mode=flpdens2d('dim',{ 'u','v','p'},'sdim',{ 'x','y'},'submode','std', ...
'tdiff','on');
appl{1}.name='ns';
appl{1}.dim={ 'u','v','p'};
appl{1}.border='off';
appl{1}.var={};
appl{1}.form='general';
appl{1}.elemdefault='Lagp2p1';
appl{1}.assign={'Fx';'Fx_ns';'Fy';'Fy_ns';'Kx';'Kx_ns';'Ky';'Ky_ns';'U'; ...
'U_ns';'V';'V_ns';'beta_x_uv';'beta_x_uv_ns';'beta_y_uv';'beta_y_uv_ns'; ...
'cBeta_uv';'cBeta_uv_ns';'eta';'eta_ns';'rho';'rho_ns';'wequ_u';'wequ_u_ns'; ...
'wequ_v';'wequ_v_ns'};
appl{1}.shape={'shlag(2,'u'),'shlag(2,'v'),'shlag(1,'p)'};
appl{1}.sshape=2;
appl{1}.equ.rho={'1'};
appl{1}.equ.eta={'1'};
appl{1}.equ.Fx={'0'};
appl{1}.equ.Fy={'0'};
appl{1}.equ.sdon={0};
appl{1}.equ.sdtype={'sdlowre'};
appl{1}.equ.delta={'0.25'};
appl{1}.equ.gporder={{4;4;2}};
appl{1}.equ.cporder={{2;2;1}};
appl{1}.equ.shape={1:3};
appl{1}.equ.init={{{'0'};{'0'};{'0'}}};
appl{1}.equ.usage={1};
appl{1}.equ.ind=1;
appl{1}.bnd.u={'0'};
appl{1}.bnd.v={'0'};
appl{1}.bnd.p={'0'};
appl{1}.bnd.type={'noslip'};
appl{1}.bnd.gporder={{0:0:0}};
appl{1}.bnd.cporder={{0:0:0}};
appl{1}.bnd.shape={0};
appl{1}.bnd.ind=ones(1,7);

```

```

% Application mode 2
appl{2}.mode=flpdeht2d('dim',{ 'T'}, 'sdim',{ 'x', 'y'}, 'submode','std', 'tdiff', ...
'on');
appl{2}.name='ht';
appl{2}.dim={ 'T'};
appl{2}.border='off';
appl{2}.var={};
appl{2}.form='coefficient';
appl{2}.elemdefault='Lag2';
appl{2}.assign={'C'; 'C_ht'; 'Const'; 'Const_ht'; 'Ctrans'; 'Ctrans_ht'; 'Q'; ...
'Q_ht'; 'Tamb'; 'Tamb_ht'; 'Tambtrans'; 'Tambtrans_ht'; 'Text'; 'Text_ht'; 'Tinf'; ...
'Tinf_ht'; 'flux'; 'flux_ht'; 'flux_x'; 'flux_x_ht'; 'flux_y'; 'flux_y_ht'; ...
'gradT'; 'gradT_ht'; 'gradTx'; 'gradTx_ht'; 'gradTy'; 'gradTy_ht'; 'h'; 'h_ht'; ...
'htrans'; 'htrans_ht'; 'n_flux'; 'n_flux_ht'; 'q'; 'q_ht'; 'rho'; 'rho_ht'};
appl{2}.shape={ 'shlag(2, "T")'};
appl{2}.sshape=2;
appl{2}.equ.rho={ '1'};
appl{2}.equ.C={ '1'};
appl{2}.equ.k={{ '1'}};
appl{2}.equ.Q={ '1'};
appl{2}.equ.htrans={ '0'};
appl{2}.equ.Text={ '0'};
appl{2}.equ.Ctrans={ '0'};
appl{2}.equ.Tambtrans={ '0'};
appl{2}.equ.gporder={{ 4}};
appl{2}.equ.cporder={{ 2}};
appl{2}.equ.shape={ 1};
appl{2}.equ.init={{ '0'}};
appl{2}.equ.usage={ 1};
appl{2}.equ.ind=1;
appl{2}.bnd.q={ '0'};
appl{2}.bnd.h={ '0'};
appl{2}.bnd.Tinf={ '0'};
appl{2}.bnd.Const={ '0'};
appl{2}.bnd.Tamb={ '0'};
appl{2}.bnd.T={ '0'};
appl{2}.bnd.type={ 'T'};
appl{2}.bnd.gporder={{ 0}};
appl{2}.bnd.cporder={{ 0}};
appl{2}.bnd.shape={ 0};
appl{2}.bnd.ind=ones(1,7);
fem.appl=appl;

```

```

% Initialize mesh
fem.mesh=meshinit(fem,...
'Out', {'mesh'},...
'jiggle', 'mean',...
'Hcurve', 0.2999999999999999,...
'Hgrad', 1.3,...
'Hpnt', {10,[]});
% Differentiation rules
fem.rules={};
% Problem form
fem.outform='general';
% Differentiation simplification
fem.simplify='on';
% Boundary conditions
clear bnd
bnd.u={'0','0','0','0'};
bnd.v={'0','v0','0','0'};
bnd.p={'0','0','0','0'};
bnd.type={'slip','uv','out','noslip'};
bnd.gporder={{0:0:0},{0:0:0},{0:0:0},{0:0:0}};
bnd.cporder={{0:0:0},{0:0:0},{0:0:0},{0:0:0}};
bnd.shape={0,0,0,0};
bnd.ind=[1 2 1 3 1 4 4];
fem.appl{1}.bnd=bnd;
% Boundary conditions
clear bnd
bnd.q={'0','0','0'};
bnd.h={'0','0','0'};
bnd.Tinf={'0','0','0'};
bnd.Const={'0','0','0'};
bnd.Tamb={'0','0','0'};
bnd.T={'0','T0','T1'};
bnd.type={'q0','T','T'};
bnd.gporder={{0},{0},{0}};
bnd.cporder={{0},{0},{0}};
bnd.shape={0,0,0};
bnd.ind=[1 2 1 1 1 3 3];
fem.appl{2}.bnd=bnd;

```

```

% PDE coefficients
clear equ
equ.rho={'rho0'};
equ.eta={'mu'};
equ.Fx={'0'};
equ.Fy={'alpha0*g0*rho0*(T-T0)'};
equ.sdon={0};
equ.sdtype={'sdlowre'};
equ.delta={'0.25'};
equ.gporder={{4;4;2}};
equ.cporder={{2;2;1}};
equ.shape={1:3};
equ.init={{{'0'}};{'v0'}};{'0'}};
equ.usage={1};
equ.ind=1;
fem.appl{1}.equ=equ;
% PDE coefficients
clear equ
equ.rho={'rho0'};
equ.C={'cp'};
equ.k={{{'kc'}}};
equ.Q={'-cp*rho0*(Tx*u+Ty*v)'};
equ.htrans={'0'};
equ.Text={'0'};
equ.Ctrans={'0'};
equ.Tambtrans={'0'};
equ.gporder={{4}};
equ.cporder={{2}};
equ.shape={1};
equ.init={{{'T0'}}};
equ.usage={1};
equ.ind=1;
fem.appl{2}.equ=equ;

```

```
% Internal borders
fem.appl{1}.border='off';
% Internal borders
fem.appl{2}.border='off';
% Shape functions
fem.appl{1}.shape={'shlag(2,'u'),'shlag(2,'v'),'shlag(1,'p)'};
% Shape functions
fem.appl{2}.shape={'shlag(2,'T')'};
% Geometry element order
fem.appl{1}.sshape=2;
% Geometry element order
fem.appl{2}.sshape=2;
% Define constants
fem.const={...
'rho0', 1000,...
'mu', 0.001,...
'cp', 4200,...
'kc', 0.59999999999999998,...
'alpha0', 0.00018000000000000001,...
'g0', 9.8000000000000007,...
'v0', 0.0050000000000000001,...
'T0', 293,...
'T1', 303};
```

```
% Multiphysics
fem=multiphysics(fem);
% Extend the mesh
fem.xmesh=meshextend(fem,'context','local','cplbndeq','on','cplbndsh','on');
% Evaluate initial condition
init=asseminit(fem,...
'context','local',...
'init', fem.xmesh.elemin);
% Solve nonlinear problem
fem.sol=femnlin(fem,...
'out', 'sol',...
'stop', 'on',...
'init', init,...
'report', 'on',...
'context','local',...
'sd', 'off',...
'nullfun','fnullorth',...
'blocksize',5000,...
'solcomp',{'T','p','u','v'},...
'linsolver','matlab',...
'bsteps', 0,...
'ntol', 9.9999999999999995e-007,...
'hnlm', 'off',...
'jacobian','equ',...
'maxiter',25,...
'method', 'eliminate',...
'uscale', 'auto');
```

- % Save current fem structure for restart purposes
- fem0=fem;
- % Plot solution
- postplot(fem,...
- 'geomnum',1,...
- 'context','local',...
- 'tridata',{'T','cont','internal'},...
- 'trifacestyle','interp',...
- 'triedgestyle','none',...
- 'trimap', 'jet',...
- 'trimaxmin','off',...
- 'tribar', 'on',...
- 'geom', 'on',...
- 'geomcol','bginv',...
- 'refine', 3,...
- 'contorder',2,...
- 'phase', 0,...
- 'title', 'Surface: temperature (T) ',...
- 'renderer','zbuffer',...
- 'solnum', 1,...
- 'axisvisible','on')