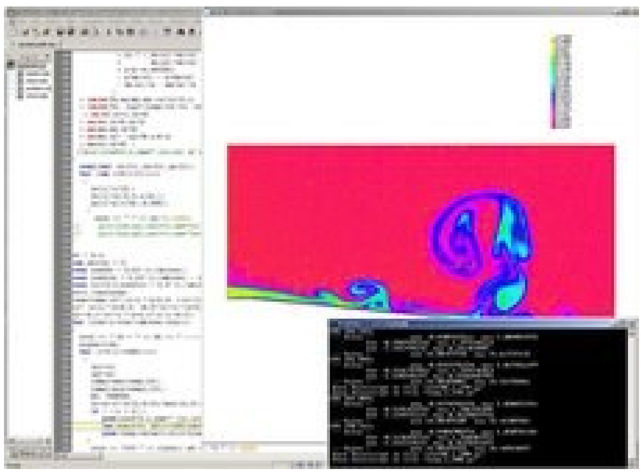
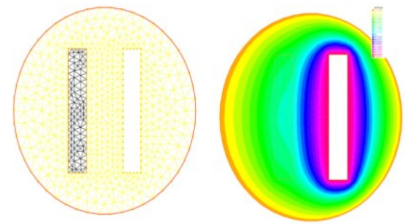
Computational Design Lab

FreeFEM ++ Tutorial for Structural Optimization

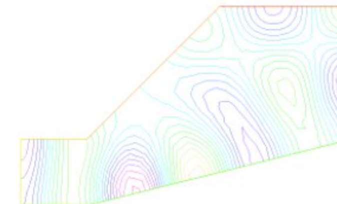
한양대학교
HANYANG UNIVERSITY

Introduction

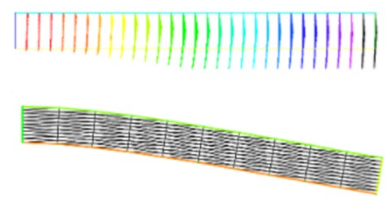




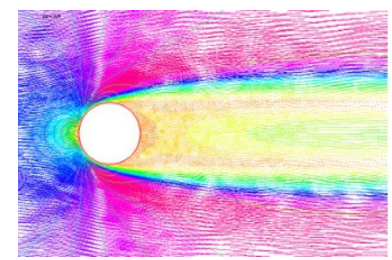
Heat Transfer



Acoustics



System of Elasticity



System of Fluid

- **1D, 2D, 3D 및 3D 비선형 다중 물리 시스템을 위한 편미분 방정식 Solver**
- 오픈 소스 코드 / 무료 배포 소프트웨어로서의 장점이 있으나 해석 결과에 대한 책임을 지지 않음
- **FreeFEM++** C++ 로 개발되었으며 작동 언어 또한 C++ 로 구성

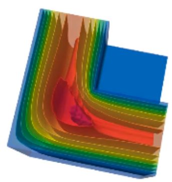
Installation

https://freefem.org/ >> DOCUMENTATION >> DOWNLOAD

FREEFEM | DOCUMENTATION | COMMUNITY | MODULES | SOURCE CODE | GALLERY | EVENTS | TRY IT ONLINE | DONATE

FREEFEM DAYS
12TH EDITION - PARIS

```
load "msh3"

// Parameters
int nn = 20; // Mesh quality

// Mesh
int[int] labs = [1, 2, 2, 1, 1, 2]; // Label numbering
mesh3 Th = cube(nn, nn, nn, label=labs);
// Remove the ]0.5,1[^3 domain of the cube
Th = trunc(Th, (x < 0.5) | (y < 0.5) | (z < 0.5), label=1

// Fespace
fespace Vh(Th, P1);
Vh u, v;

// Macro
macro Grad(u) [dx(u), dy(u), dz(u)] //

// Define the weak form and solve
solve Poisson(u, v, solver=CG)
    = int3d(Th)(
          Grad(u)' * Grad(v)
    )
    -int3d(Th)(
          1 * v
    )
    + on(1, u=0)
    ;

// Plot
plot(u, nbiso=15);
```



A high level multiphysics finite element software

FreeFEM offers a fast interpolation algorithm and a language for the manipulation of data on multiple meshes.

v4.8
Release notes

Download

All

Download

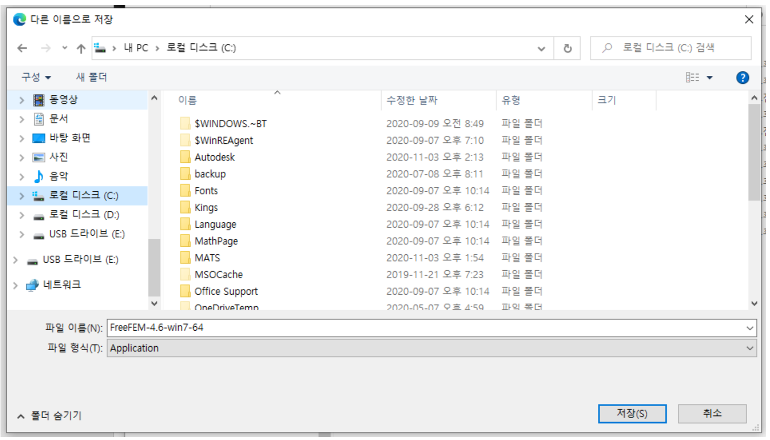
Latest binary packages

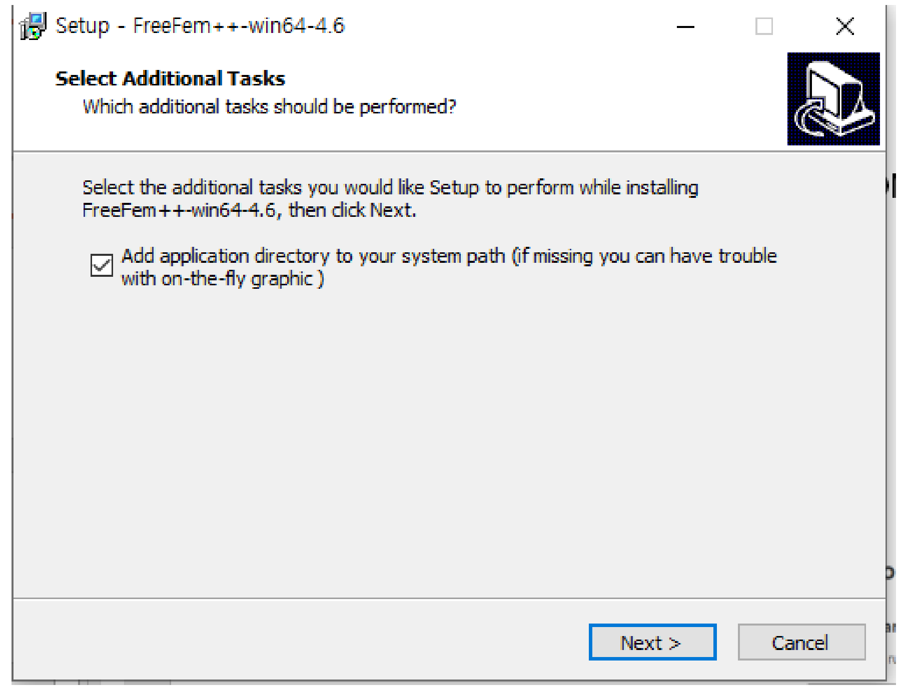
FreeFEM v4.6 runs under macOS, Ubuntu, and 64-bit Windows.

Operating System	FreeFEM Version	Size	Date
macOS 10.10.5 or higher	4.5	412 MB	Feb 11, 2020
Ubuntu 16.04 or higher	4.6	212 MB	Mar 02, 2020
64-bit Windows | 4.6 | 185 MB			Mar 02, 2020
Docker image	4.6	487 MB	Mar 02, 2020
Source 4.6	4.6	12.4 MB	Mar 02, 2020
previous releases	–	–	–

Installation

Setup file >> Select Additional Tasks >> Next >> Install





Program 설치 path에 한글이 포함되지 않도록 주의

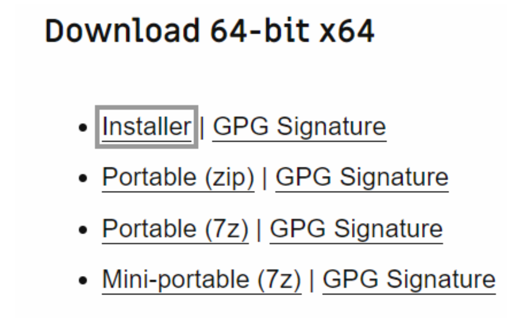
Ex . (C:\Users\Documents**프로그램**\FreeFem++)

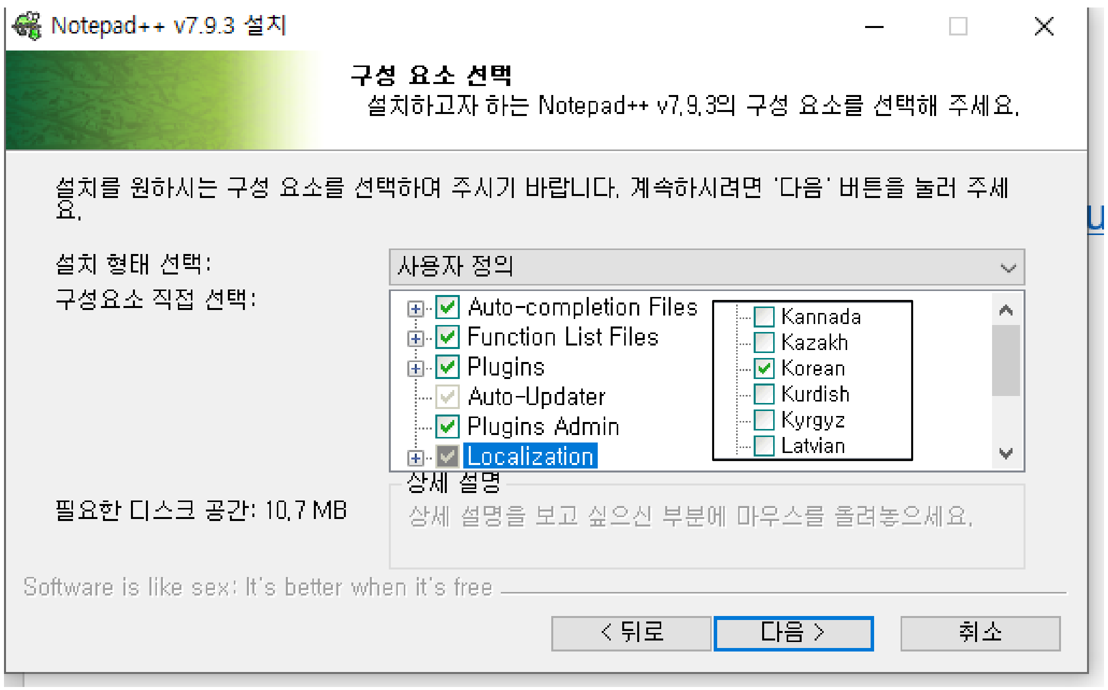
C++ Editor (Notepad++)

FreeFEM++의 Input code 편집기로 사용

Notepad++ Download

(https://notepad-plus-plus.org/downloads/v7.9.3/)



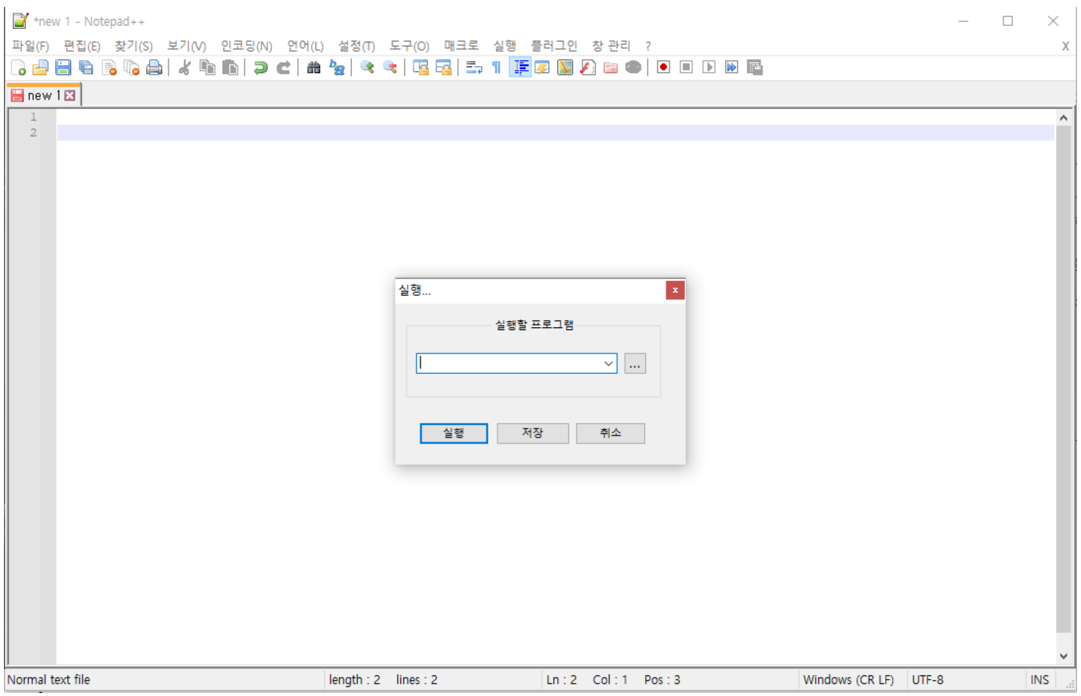


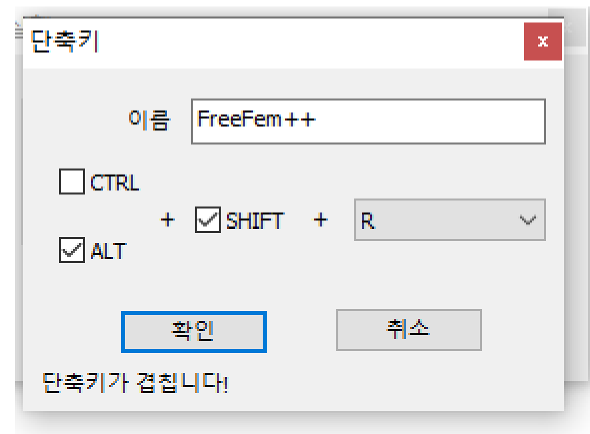
언어 한글 추가

Localization >> Korean

Notepad++ FreeFem ++ 실행 설정

Notepad ++ 실행 >> 단축키 “enter” + “F5” >> 클릭 “…”
>>In FreeFEM++ 경로의 “launch++” 파일 선택 후 “저장”
>>FreeFem++를 실행할 단축키 설정 (ex.”alt”+”shift”+”R”)



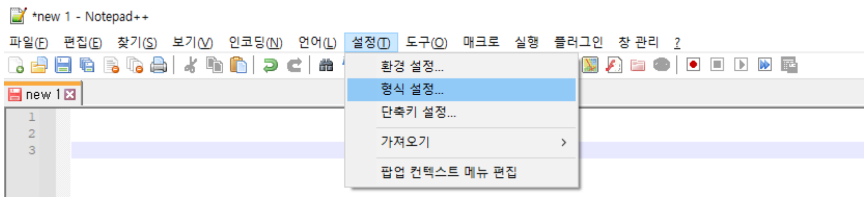


Notepad++ 창에서 단축키를 이용해 바로 FreeFem++ 실행가능

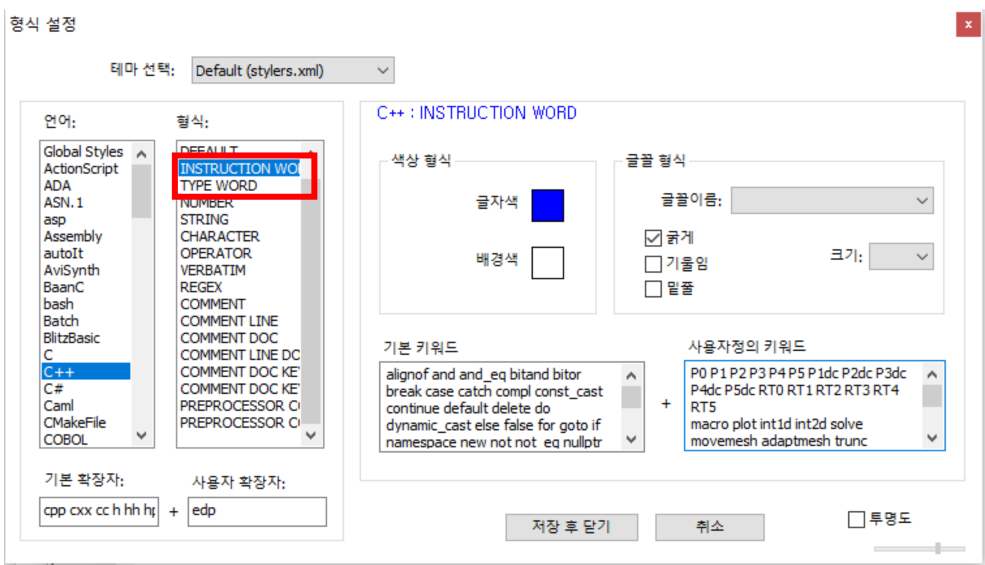
Notepad++ FreeFem 형식 설정 추가

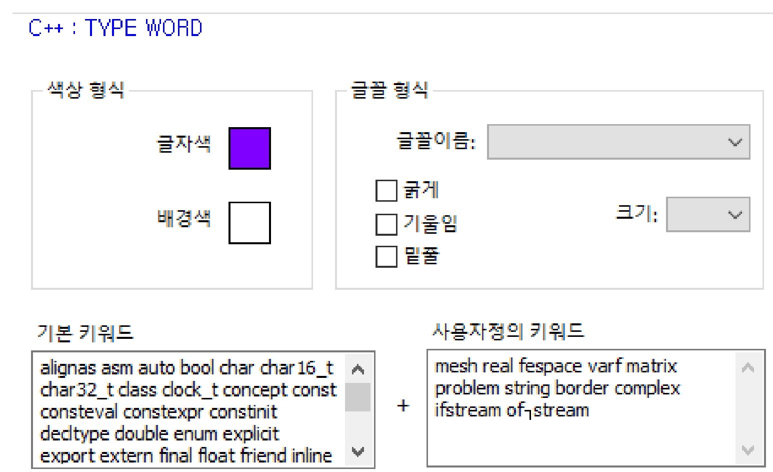
형식 설정 : FreeFEM 코드 텍스트를 형식에 따라 색을 부여

설정 >> 형식설정



언어/형식/사용자확장자/사용자정의 키워드(형식 Instruction word, Type word) (http://www3.freefem.org/ff++/color-syntax-win.pdf)에 나와있는 설정 및 키워드 저장 후 닫기

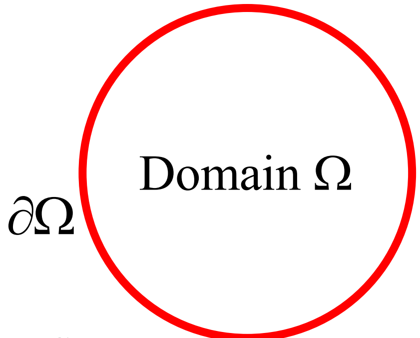




실행 예제

FreeFEM++ 가이드에 다양한 예제 제공
(http://www3.freefem.org/ff++/color-syntax-win.pdf)
In page.23, 2-D PDE: Poisson

Notepad++ 에서 새파일 >> 다른이름으로 저장(*.edp)



Boundary $\partial\Omega = C$

$$C = \left[(x,y) \mid x = \cos(t), y = \sin(t), 0 \le t \le 2\pi \right]$$

$$-\Delta u(x,y) = f(x,y) \quad \text{in} \quad \Omega$$

$$\Delta u(x,y) = 0 \qquad \text{on } \partial\Omega$$

$$f(x,y) = xy$$

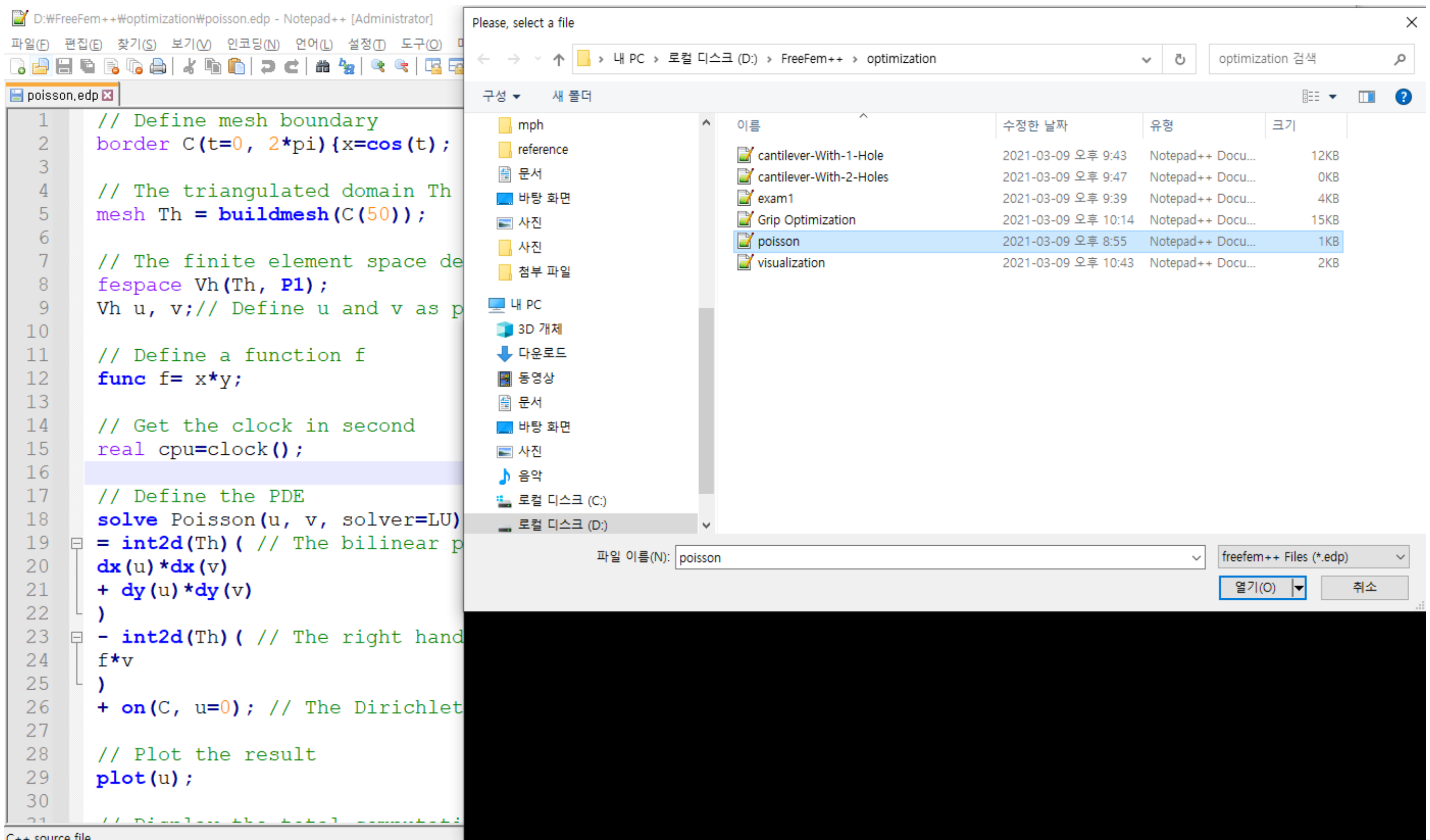
지배방정식은
Variational formulation으로 입력

$$\int_{T_h} \left(\frac{\partial u}{\partial x}\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\frac{\partial v}{\partial y} \right) dxdy = \int_{T_h} fv dxdy$$

```
// Define mesh boundary
border C(t=0, 2*pi){x=cos(t); y=sin(t);}

// The triangulated domain Th is on the left side of its boundary
mesh Th = buildmesh(C(50));

// The finite element space defined over Th is called here Vh
fespace Vh(Th, P1);
Vh u, v;// Define u and v as piecewise-P1 continuous functions

// Define a function f
func f= x*y;

// Get the clock in second
real cpu=clock();

// Define the PDE
solve Poisson(u, v, solver=LU)
= int2d(Th)( // The bilinear part
dx(u)*dx(v)
+ dy(u)*dy(v)
)
- int2d(Th)( // The right hand side

f*v
)
+ on(C, u=0); // The Dirichlet boundary condition

// Plot the result
plot(u);

// Display the total computational time
cout << "CPU time = " << (clock()-cpu) << endl;
```

FreeFEM++ 실행

Notepad ++새문서
>>파일명.edp 저장 (파일형식 all type)
>>input text 붙여넣기
>>FreeFem++ 실행 (실행 단축키)

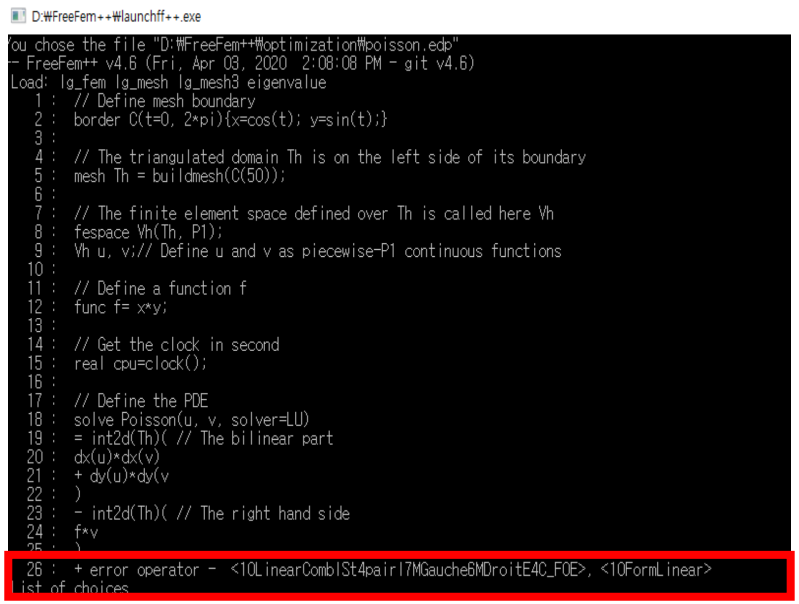


FreeFEM++ 디버깅

프로그래밍 중 발생하는 논리적인 오류나 오타로 비정상적 연산이 발생할 때 디버깅 수행

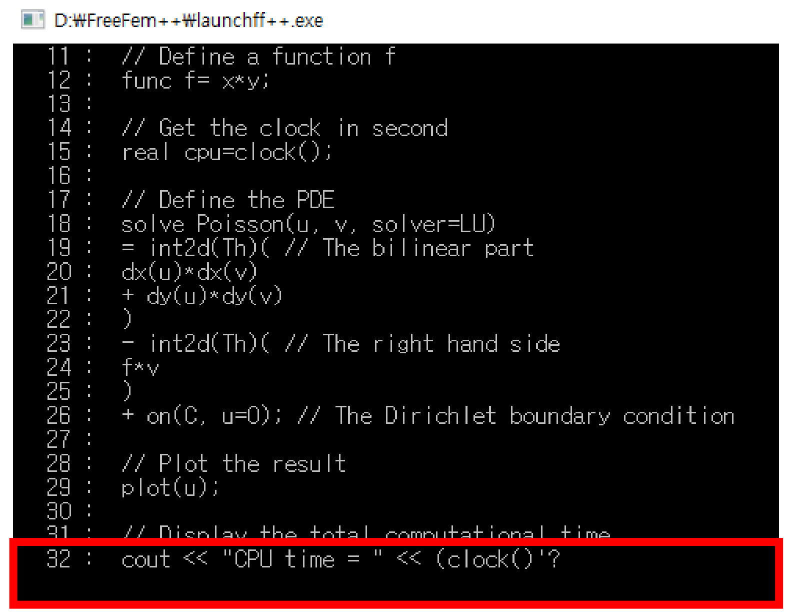
흔히 발견되는 오류 출력 유형

1. 오타로 인한 에러 메시지 출력



```
D:₩FreeFem++₩launchff++.exe
You chose the file "D:₩FreeFem++₩optimization₩poisson.edp"
-- FreeFem++ v4.6 (Fri, Apr 03, 2020  2:08:08 PM - git v4.6)
Load: lg_fem lg_mesh lg_mesh3 eigenvalue
    1 :  // Define mesh boundary
    2 :  border C(t=0, 2*pi){x=cos(t); y=sin(t);}
    3 :
    4 :  // The triangulated domain Th is on the left side of its boundary
    5 :  mesh Th = buildmesh(C(50));
    6 :
    7 :  // The finite element space defined over Th is called here Vh
    8 :  fespace Vh(Th, P1);
    9 :  Vh u, v;// Define u and v as piecewise-P1 continuous functions
   10 :
   11 :  // Define a function f
   12 :  func f= x*y;
   13 :
   14 :  // Get the clock in second
   15 :  real cpu=clock();
   16 :
   17 :  // Define the PDE
   18 :  solve Poisson(u, v, solver=LU)
   19 :  = int2d(Th)( // The bilinear part
   20 :  dx(u)*dx(v)
   21 :  + dy(u)*dy(v
   22 :  )
   23 :  - int2d(Th)( // The right hand side
   24 :  f*v
   25 :  )
   26 :  + error operator -  <10LinearCombISt4pairI7MGauche6MDroitE4C_F0E>, <10FormLinear>
 List of choices
```

2. 논리 오류 등으로 인한 동작 중지 후 “?” 출력

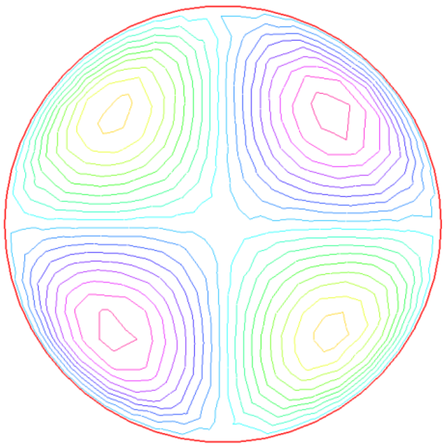


```
D:₩FreeFem++₩launchff++.exe
 11 :  // Define a function f
 12 :  func f= x*y;
 13 :
 14 :  // Get the clock in second
 15 :  real cpu=clock();
 16 :
 17 :  // Define the PDE
 18 :  solve Poisson(u, v, solver=LU)
 19 :  = int2d(Th)( // The bilinear part
 20 :  dx(u)*dx(v)
 21 :  + dy(u)*dy(v)
 22 :  )
 23 :  - int2d(Th)( // The right hand side
 24 :  f*v
 25 :  )
 26 :  + on(C, u=0); // The Dirichlet boundary condition
 27 :
 28 :  // Plot the result
 29 :  plot(u);
 30 :
 31 :  // Display the total computational time
 32 :  cout << "CPU time = " << (clock()'?
```

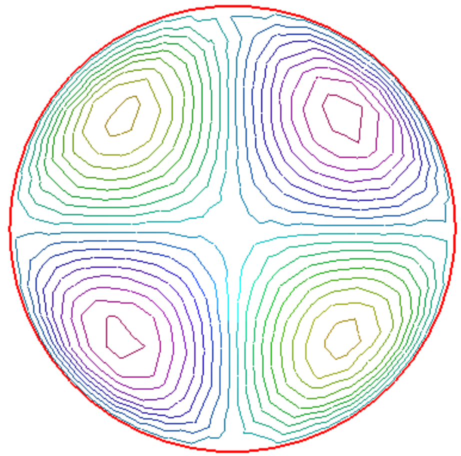
문자열 번호 확인 후 해당 코드 열에 대한 디버깅 수행

Plot Option

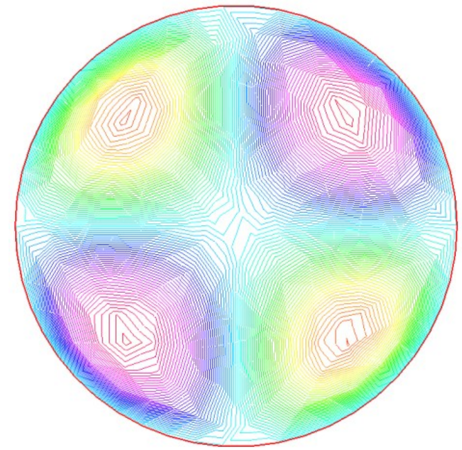
FreeFEM++ Plot 창에서 “?”(=shift+/) 키를 누르면 plot option을 확인할 수 있음



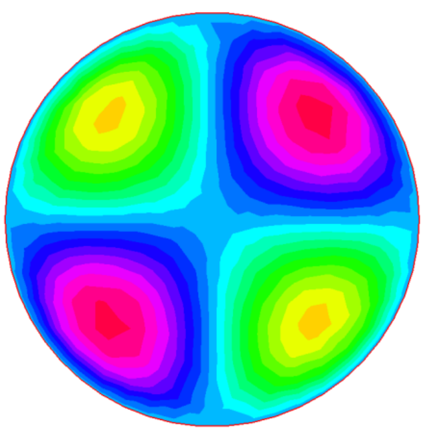
L : Light on/off



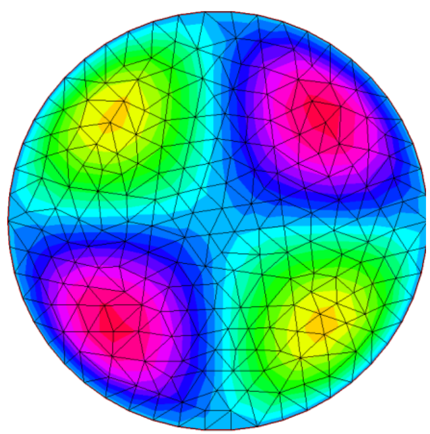
N, Shift +N : Color range



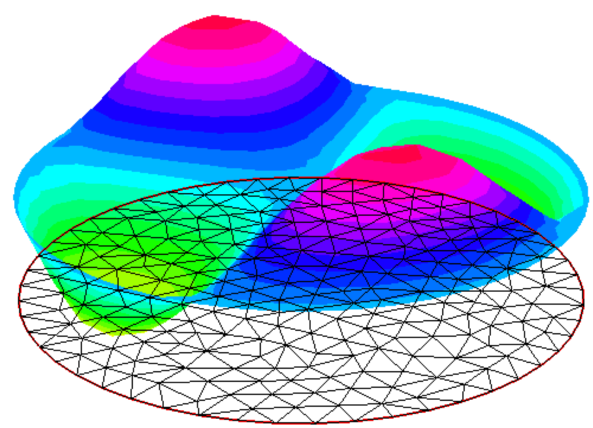
F : Filled Contour



M : Mesh plot



3 : 3D view



Structural Optimization using FreeFEM++

Struct Multidisc Optim (2006) 32: 173–181
DOI 10.1007/s00158-006-0017-y

EDUCATIONAL ARTICLE

G. Allaire · O. Pantz

Structural optimization with `FreeFem++`

Received: 23 June 2005 / Revised manuscript received: 7 September 2005 / Published online: 8 July 2006

Abstract The aim of this paper is to show that relatively small, simple, and efficient shape optimization routines can be written using the free finite element software `FreeFem++`. This is illustrated by the implementation of two classical methods: the boundary variation method and the homogenization one. Even though these routines are simple enough so that their implementation can be assigned (partially or totally) as homework to graduate students, they yield results accurate enough to be useful tools for engineers or researchers.

Keywords Shape and topology optimization · Finite element software

해당 링크에서 FreeFEM++ input code 제공
(http://www.cmap.polytechnique.fr/~optopo/OptFree/)

Index of /~optopo/OptFree

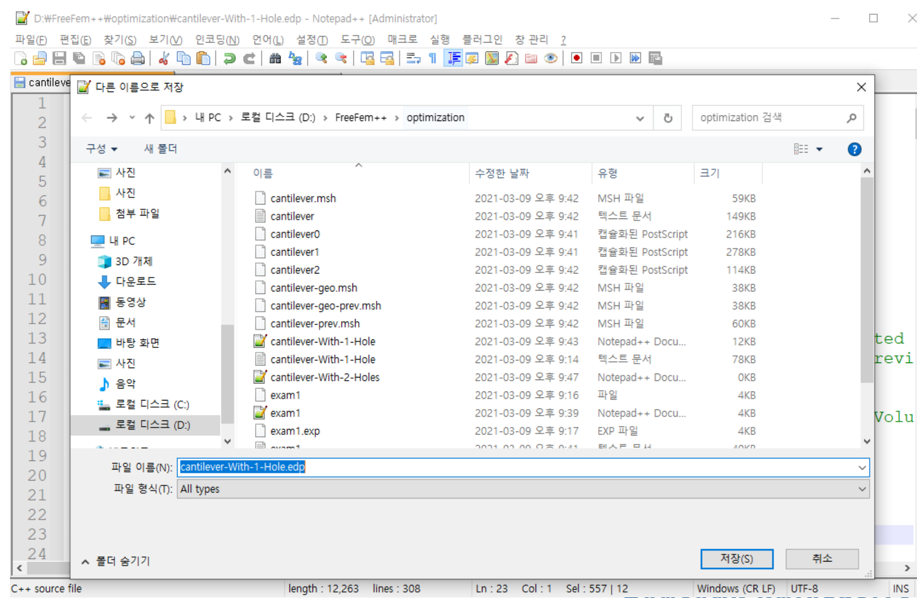
Name	Last modified	Size	Description
Parent Directory		-	
boundary-variation/	07-Sep-2005 11:10	-	
homogenization/	07-Sep-2005 11:10	-	
readme	07-Sep-2005 15:16	460	

Boundary variation method

Name	Last modified	Size	Description
Parent Directory		-	
deformed-grip.eps	07-Sep-2005 10:02	184K	
grip-geo-prev.msh	07-Sep-2005 10:02	14K	
grip-geo.msh	07-Sep-2005 10:02	14K	
grip-new.edp	07-Sep-2005 10:02	14K	
grip-prev.msh	07-Sep-2005 10:02	53K	
grip.edp	07-Sep-2005 10:02	14K	
grip.msh	07-Sep-2005 10:02	211K	
grip.txt	07-Sep-2005 10:02	57K	
grip0.eps	07-Sep-2005 10:02	44K	
grip1.eps	07-Sep-2005 10:02	255K	
grip2.eps	07-Sep-2005 10:02	75K	
reference-grip.eps	07-Sep-2005 10:02	180K	
visualization.edp	07-Sep-2005 10:02	1.5K	

링크 접속
>> “boundary-variation”
>> Compliance 또는 Grip (본 설명은 Grip 예제)
>> “{예제이름}.edp” 선택 (ex. “grip-new.edp”)
>>input text 복사

Notepad ++새문서
>>파일명.edp 저장 (파일형식 all type)
>>input text 붙여넣기
>>FreeFem++ 단축키
>>Input file 실행



Implementation (Grip)

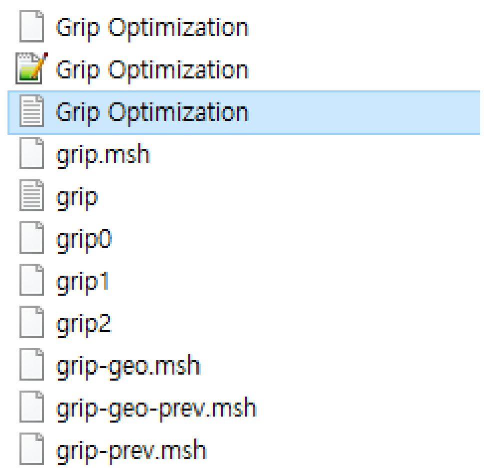
FreeFEM++ 동작 예시

```
/////////////////////////////////////////////////////
// Grip Optimization                              //
// by the boundary variation method               //
// Copyright O.Pantz, G. Allaire (2005)           //
/////////////////////////////////////////////////////

// Optimization Parameters
int niter=40;               //Number of iterations
real errelas=0.005;         //Error for the resolution of the variational problems
real metricmesh=10.;        //Metric used for the coarse mesh: Bigger=finer mesh
real initialstep=1.;            //Initial step
real maxstep=3.;            //Maximum step (<0 => no limit)
real lagrangestep=3.;           //Updating step of the Lagrange multiplier associated to the Volume's constraint
bool tocontinue=false;          //false=start a new computation, true=continue a previous computation
string backupname="grip";       //Name under which the results are stored
int nsave=1;                //Number of iterations to be saved
real volume0=-1.;           //=0 Constant Volume during the iterations, >0 Target Volume, <0 No Volume's constraint
real penVolume=0.05;            //Volume penalization
func cutVolume=(x<5.0);

// construction of the initial mesh
    mesh Sh,Th,Shprev,Thprev;
    //Boundaries with (non null)-Neumann and Dirichlet conditions are not optimized
    int neumann=1; int dirichlet=2; int teeth=4;
    //Boundaries with free conditions are optimized
    int free=3;

    border a(t=3,-3){x=0;y=t;label=free;};          //Left Boundary (Optimized Boundary)
    border b1(t=0,1){x=t;y=-3;label=neumann;};      //Low Boundary (with Neumann of Free Conditions)
    border c1(t=0,1){x=1+5*t;y=-3+2.5*t;label=free;};
    border dd1(t=6,5){x=t;y=-0.5;label=teeth;};     //Teeth of the grip
    border e(t=-0.5,0.5){x=5;y=t;label=free;};
    border dd2(t=5,6){x=t;y=0.5;label=teeth;};
    border c2(t=1,0){x=1+5*t;y=3-2.5*t;label=free;};    //Upper Boundary  (with Neumann of Free Conditions)
    border b2(t=1,0){x=t;y=3;label=neumann;};
    //A Hole with Dirichlet conditions (not Optimized)
    border GammaD(t=0,2*pi){x=cos(t)*0.2+0;y=sin(t)*0.2;label=dirichlet;};
    //Optimized Holes
    border Hole1(t=0,2*pi){x=cos(t)*0.5+2.;y=sin(t)*0.5+1.5;label=free;};
    border Hole2(t=0,2*pi){x=cos(t)*0.5+2.;y=sin(t)*0.5-1.5;label=free;};
```

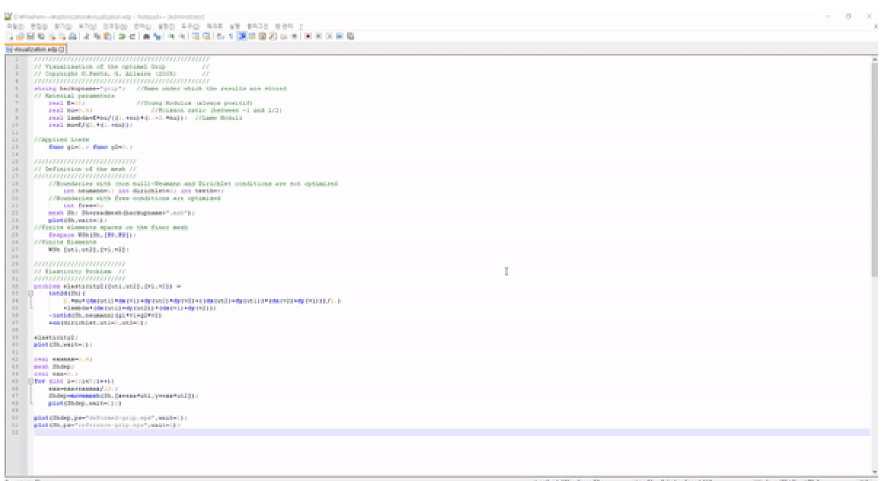
Output data

프로그래밍에 따른 Mesh와 Variable 파일 출력

```
//We save the meshes
    savemesh(Sh,backupname+".msh");
    savemesh(Shprev,backupname+"-prev.msh");
    savemesh(Th,backupname+"-geo.msh");
    savemesh(Th,backupname+"-geo-prev.msh");
//We save all variables
    {ofstream f(backupname+".txt");
    f << volume0<<"\n"; //Initial Volume
    f <<lagrange<<"\n"; //Lagrange Multiplier
    f <<actualstep<<"\n";   //Actual step
    f <<rollback<<"\n"; //boolean variable
    f <<iter<<"\n";     //Current Iteration
    f <<nsaved+1<<"\n"; //Number of back-up done
    f <<isoval<<"\n";
    [Dprev1,Dprev2]=[Dprev1,Dprev2];
    f <<Dprev1[]<<endl;f<<Dprev2[]<<endl;};
```



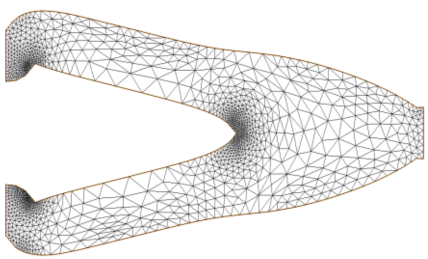
Grip 폴더>> Visualization.edp로 최적 모델의 변형을 도식화



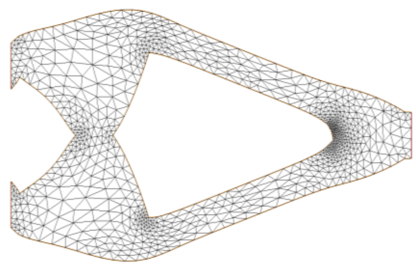
Next : Enter
Previous : P

Cantilever optimization results

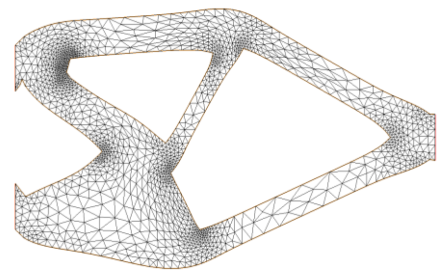
No hole
(boundary variation)



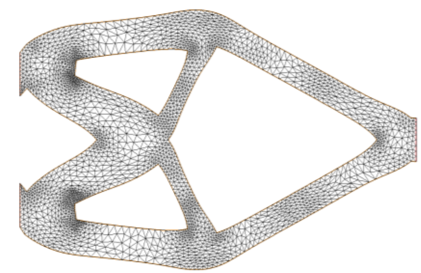
1 hole
(boundary variation)



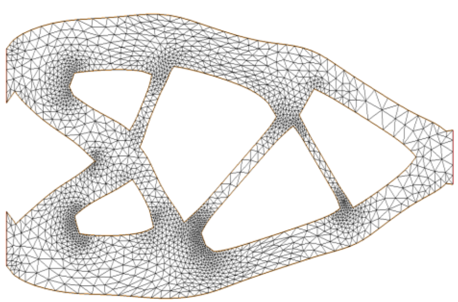
2 holes
(boundary variation)



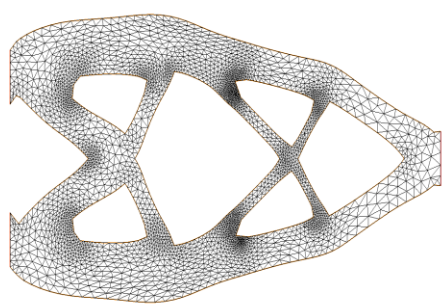
3 holes
(boundary variation)



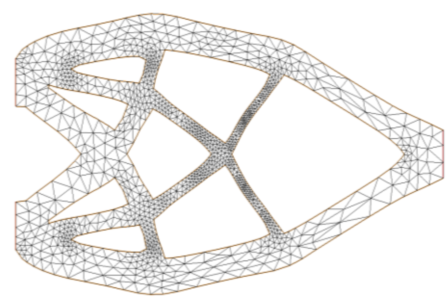
5 holes
(boundary variation)



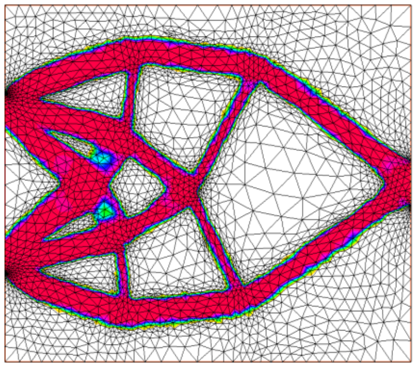
6 holes
(boundary variation)



8 holes
(boundary variation)



No hole
(Homogenization)



Thank you for your attention

Computational Design Lab