Personal Programming Project 2020-21

ISO-GEOMETRIC ANALYSIS BASED TOPOLOGY OPTIMIZATION (IGTO)

Viswambhar Reddy Yasa

Computational Material Science

65074

1 User manual

The following steps should be performed to run the program and test cases. All the files are written in python.

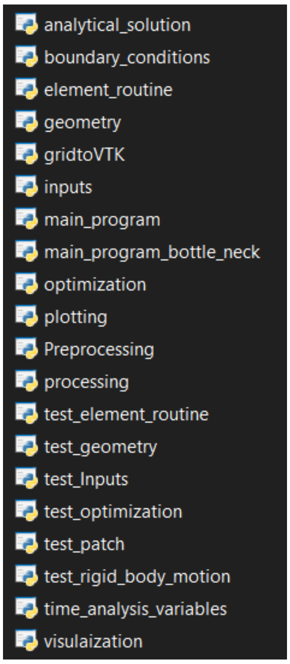

Figure 1: List of all python files

1.1 Procedure to run the program

All python files have to be placed in same folder and working directory has to be same to run the python program. Any python environment can be used to run this python code.

1. A file name main_program.py is the starting point of the IGTO program.
   command: **python main_program.py**

   (a) Required python external libraries like 'numpy', 'matplotlib', 'pyvista', 'pyEVTK','pytest' are checked and installed.

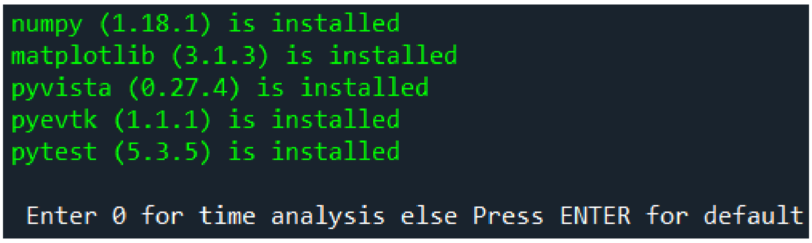

Figure 2: Installed python libraries

2. A prompt appears which asks the user to choose if time analysis has to be performed or not.
   0- Time analysis (log files are generated)
   Enter - For normal execution without any log files.

3. Then the inputs have to be given or press enter to run default values.

```
l h w nx ny nz load volume_fra penal rmin E v density BC_op Ft_op
    verbose
8 5 1 35 25 3 -100 0.4 3 1.5 150000 0.3 7850 0 1 1
INPUTS WITH DEFAULT VALUES:
Length(l)                   : 8
Height(h)                   : 5
Width(w)                    : 1
nx(no of elements along x)  : 35
ny(no of elements along y)  : 25
nz(no of elements along z)  : 3
load                        : -100
volume_fraction             : 0.4
penal                       : 3
rmin                        : 1.5
```

```
E(Youngs modulus)            : 150000
v(poisson ratio)             : 0.3
density                      : 7850
BC_op                        : 0
Ft_op                        : 1
verbose                      : 1
```

Boundary condition option (BC_op):

```
Option :
0- Cantilever beam with load along the bottom edge of the free end.
1- Simple supported with point load at bottom center.
2- Cantilever beam with point load at bottom of the free end.
3- Cantilever beam with point load at the free end (2d case
   loading at y=height and x=length).
4- Cantilever beam with two forces at top and bottom of the free
   end .
5- Cantilever beam with point load at the center of the free end.
6- Simple supported with load at top center.
```

Optimizer option (Ft_op):

```
Option :
0-OC (optimality criterion)

1-MMA (method of moving asymptotes)
```

Verbose:

```
Option :
0-Will not print plots using pyvista only VTK file is generated.

1- Plots are generated and stored in results folder.
```

4. Plots are stored in respective sub folder with optimizer names in results folder.

5. A copy of input values and time log file are stored in log_files folder.

1.2 Procedure to run the test case

We used pytest to perform unit, functional and patch testing. All test_ files are the python files which contain respective test cases.
All files should be placed in the same folder and the working directory has to be same, so that test case can run.

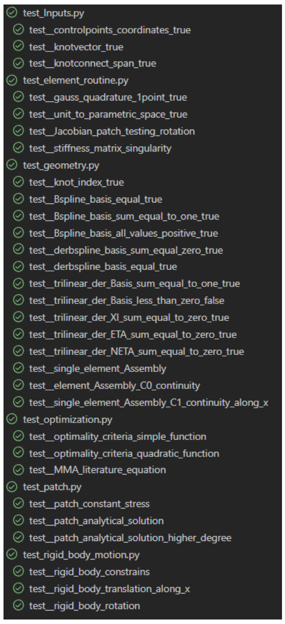

Figure 3: List of test cases

1. test_Inputs.py : Test cases for input parameters
2. test_geometry.py : Test cases on shape function and assembly
3. test_element_routine.py : Test cases on integration scheme and sanity checks on other element.
4. test_optimization.py : Test cases on optimizer function OC(optimality criterion) and MMA(method of moving asymptotes)
5. test_rigid_body_motion.py : Test cases to validate global stiffness matrix and boundary conditions by performing rigid body translation and rotation.
6. test_patch.py : Test cases on constant stress patch test and comparing analytical solution with numerical for lower and higher order shape functions.

To run all the test cases, copy all test files in same folder and enter **PYTEST** command on the terminal.
Test commands to run the respective files.

```
pytest test_Inputs.py
pytest test_geometry.py
pytest test_element_routine.py
pytest test_optimization.py
pytest test_patch.py
pytest test_rigid_body_motion.py
```

2 RESULTS VALIDATION

The input parameters are Length 8cm, Height 5cm, Width 1cm, Young's modulus 150000 N/m, Poisson ratio 0.3, load 100 N, volume fraction 0.4(optimized volume V/original volume V0), penalization factor 3, minimum radius 1.5 are same for all test cases only the boundary conditions change.
Degree of the curve
p=1,q=1,r=1.

2.1 Validation case-1

2.1.1 Problem description

Topology optimization of cantilever beam which is subjected to load at the free end. As shown in figure -4

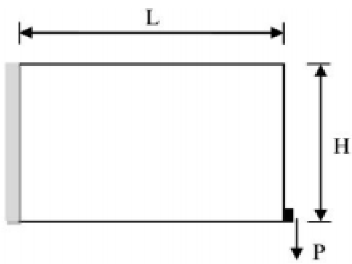

Figure 4:
Cantilever beam with load at free end

2.1.2 Results

see PPP document section 8.1

1.**MMA optimizer**

```
INPUTS:
l h w nx ny nz load volume_fra penal rmin E v density BC_op Ft_op
    verbose
8 5 1 35 25 3 -100 0.4 3 1.5 150000 0.35 7850 0 1 1
```

Figure 5: IGTO validation-1 for MMA
Results from literature

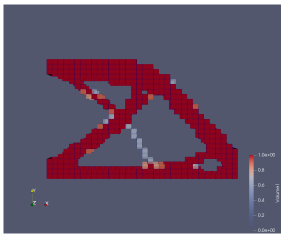

Figure 6: Optimized structure using MMA method
Program generated result, plotted in paraview

2.**OC optimizer**

```
INPUTS:
l h w nx ny nz load volume_fra penal rmin E v density BC_op Ft_op
    verbose
8 5 1 35 25 3 -100 0.4 3 1.5 150000 0.35 7850 0 0 1
```

Figure 7: IGTO validation-1 for OC
Results from literature

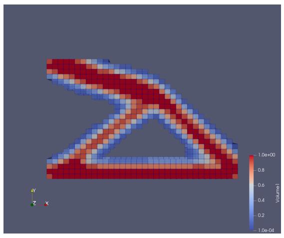

Figure 8: Optimized structure using OC method
Program generated result, plotted in paraview

2.2 Validation case-2

see PPP document section 8.2

2.2.1 Problem description

Topology optimization of cantilever beam which is subjected to load acting at the center of the free end. As shown in figure -9

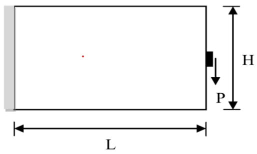

Figure 9:
Cantilever beam with load at the center of the free end

2.2.2 Results

1.**MMA optimizer**

```
INPUTS:
l h w nx ny nz load volume_fra penal rmin E v density BC_op Ft_op
    verbose
8 5 1 35 25 3 -100 0.4 3 1.5 150000 0.35 7850 0 1 1
```

Figure 10: IGTO validation-1 for MMA
Results from literature

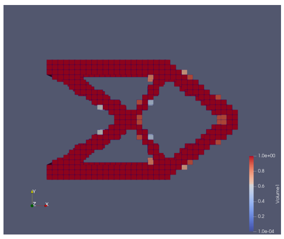

Figure 11: Optimized structure using MMA method
Program generated result, plotted in paraview

2. **OC optimizer**

```
INPUTS:
l h w nx ny nz load volume_fra penal rmin E v density BC_op Ft_op
   verbose
8 5 1 35 25 3 -100 0.4 3 1.5 150000 0.35 7850 0 0 1
```

Figure 12: IGTO validation-2 for OC
Results from literature

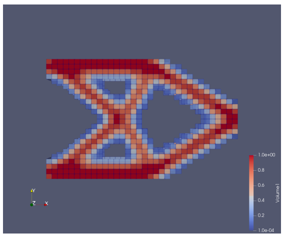

Figure 13: Optimized structure using OC method
Program generated result, plotted in paraview

2.3 Validation case-3

2.3.1 Problem description

Topology optimization of simple supported beam which is subjected to load acting at the center. As shown in figure -14

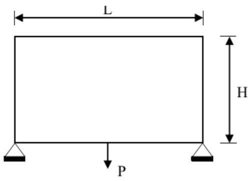

Figure 14:
Simple supported beam with load at the center of the free end

2.3.2 Results

see PPP document section 8.3
1.**MMA optimizer**

```
INPUTS:
l h w nx ny nz load volume_fra penal rmin E v density BC_op Ft_op
    verbose
8 5 1 35 25 3 -100 0.4 3 1.5 150000 0.35 7850 0 1 1
```

Figure 15: IGTO validation-3 for MMA
Results from literature

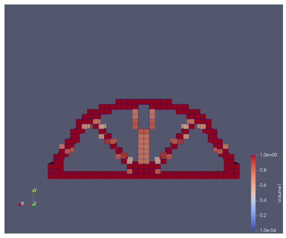

Figure 16: Optimized structure using MMA method
Program generated result, plotted in paraview

2. **OC optimizer**

```
INPUTS:
l h w nx ny nz load volume_fra penal rmin E v density BC_op Ft_op
    verbose
8 5 1 35 25 3 -100 0.4 3 1.5 150000 0.35 7850 0 0 1
```

Figure 17: IGTO validation-2 for OC
Results from literature

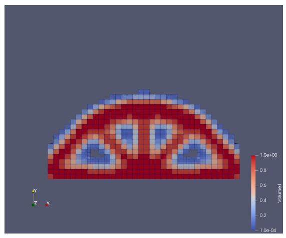

Figure 18: Optimized structure using OC method
Program generated result, plotted in paraview

NOTE:

1. Recommended to use SYPDER (ANACONDA) python IDE for better visualization of output if using PYVISTA.

2. if the program is not able to plot results, enter verbose as 0.

```
INPUTS:
l h w nx ny nz load volume_fra penal rmin E v density BC_op Ft_op
    verbose
8 5 1 35 25 3 -100 0.4 3 1.5 150000 0.35 7850 0 0 0
```

3. visualization of optimized structure can be done using the VTK file generated in paraview. For more details see *PPP document section 5.5.2*