



건물의 3차원 구조체에 대한 전열해석 프로그램 개발 중 3차원 모델의 해석 오류 저감을 위한 사전 수정 방법 연구

preprocessing methodology to reducing calculation errors in 3 dimensional model for development of heat transfer analysis program for 3 dimensional structure of building

이규성* · 이주희** · 이용준***

Lee, Kyusung* · Lee, Juhee** · Lee, Yongjun***

* Corresponding author, Doall Tech, Seoul, Korea (kslee@doalltech.com)

*** Dept. of Computer Engineering, Hoseo Univ., South Korea

**** BEL Technology CO.Ltd., Seoul, Korea

ABSTRACT

This study is part of three-dimensional(3D) heat transfer analysis program developmental process. The program is being developed without it's own built in 3D-modeller. So 3D-model must be created from another 3D-modeller such as generic CAD programs and imported to the developed program. After that, according to the 3D-geometric data form imported model, 3D-mesh created for numerical calculation. But the 3D-model created from another 3D-modeller is likely to have errors in it's geometric data such as mismatch of position between vertexes or surfaces. these errors make it difficult to create 3D-mesh for calculation. These errors are must be detected and cured in the pre-process before creating 3D-mesh. So, in this study four kinds of filters and functions are developed and tested. Firstly, 'vertex error filter' is developed for detecting and curing for position data errors between vertexes. Secondly, 'normal vector error filter' is developed for errors of surface's normal vector in 3D-model. Thirdly, 'intersection filter' is developed for extracting and creating intersection surface between adjacent objects. fourthly, 'polygon-line filter' is developed for indicating outlines of object in 3D-model. the developed filters and functions were tested on several shapes of 3D-models. and confirmed applicability. these developed filters and functions will be applied to the developed program and tested and modified continuously for less errors and more accuracy.

KEYWORD

전산해석
전열해석
비정렬격자
행렬계산
격자중심방법

Numerical methods
Heat transfer
Unstructured grid, matrix solver
Cell-centered scheme

ACCEPTANCE INFO

Received January 4, 2016
Final revision received February 12, 2016
Accepted February 16, 2016

© 2016 KIEAE Journal

1. 서론

1.1. 연구의 배경

본 연구는 3차원 정상상태 전열해석 프로그램(three dimensional steady state heat transfer program)의 국산화 개발 연구의 수행 성과 중 일부로서, 개발 중인 프로그램은 '공동주택의 결로 방지 설계 기준' 제도 시행(2014.05)에 따라 공동주택 신축 시 의무적으로 요구되는 설계안의 기준 만족 여부 판단을 위하여 수행하여야 하는 다양한 공동주택 외피(외벽 및 창호)에 대한 3차원 정상상태 전열해석을 통한 결로방지성능 평가를 목적으로 한다.

당 논문은 개발 중인 프로그램이 3차원 전열 해석을 수행하기 위한 해석 모델의 작성에 필요한 전처리 프로세스의 개발에 관한 연구 내용을 중심으로 하였으며, 전처리 프로세스의 수행을 위한 기능별 알고리즘의 개발와 이에 기초하여 작성된 프로그램의 구현 기능을 검증한 내용을 중심으로 하였다.

일반적으로 수치해석을 수행하는 프로그램은 해석 수행을 위

해서 Fig. 1 과 같이 기본적으로 전처리 과정(pre-process), 해석 과정(analysis process), 후처리 과정(post-process)으로 구성된다.

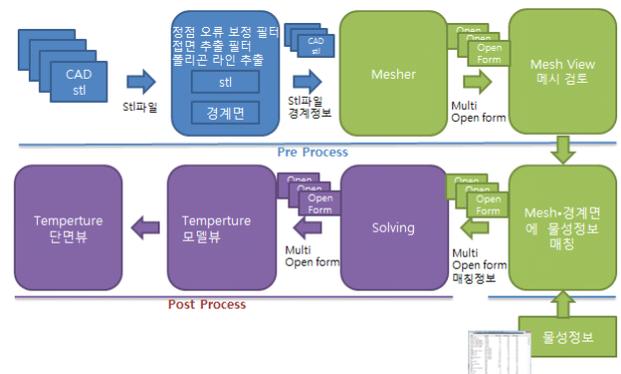


Fig. 1. heat transfer analysis Process

각 과정을 간략히 설명하면 전처리 과정은 해석 대상에 대한 형상 모델 작성, 해석 조건의 설정, 해석 방법의 설정 등 실제 수치해석을 수행하기 위한 기초 작업을 수행하는 과정과 작성된 형

상 정보를 기초로 해석을 수행하기 위한 mesh를 생성하는 과정으로 구성되며, 해석 과정(analysis process)은 전처리 과정에서 설정된 모델 및 조건에 따라 실제 해석을 수행하는 과정이고, 후처리 과정(post-process)은 해석된 결과를 텍스트 또는 그래픽적인 방법으로 출력하여 결과를 이해할 수 있도록 처리하는 과정이다.

1.2. 연구의 목적

본 연구에서는 전처리 과정 중 형상 모델을 프로그램에 불러들인 후 (import) 형상 모델에 해석 조건을 설정하고 해석을 위한 mesh를 생성할 수 있도록 모델의 형상 데이터를 전처리하는 필수 기능이다.

개발된 기능은 import된 형상 데이터에 포함되어 있을 수 있는 오류를 추출하는 기능, 서로 다른 재질의 물질이 접하는 면을 추출하여 구분하는 기능, 정확한 형상을 표현하기 위하여 재질별 외곽선을 추출하는 기능이다.

본 연구의 개발 프로그램은 3차원 형상을 직접 작성하는 모델러를 개발하지 않고 범용 CAD 프로그램에서 작성한 3차원 형상을 STL(STereoLithography) 파일로 export하여 이를 프로그램의 형상 모델로 import하는 방식으로 3차원 모델을 작성하는 방식을 선택하였기 때문에 상기의 개발 기능은 해석을 위한 모델의 작성을 위하여 필수적인 기능이다.

특히 범용 CAD 프로그램은 모델 작성 시 정점 선택(object snap)시 일정한 오차를 발생시킬 수 있으며, 이에 대한 오류 추출 및 수정 기능은 mesh 생성을 위한 필수적인 기능이다. 또한 이질 재료가 접하는 접합면의 추출은 해석 시 이질 재료간 경계조건(interface boundary)의 설정을 통한 열전도율의 차이를 반영할 수 있도록 하기 위한 필수적인 기능이다.

2. 열전도 해석을 위한 표면 전처리 기능 개발

2.1. 표면 전처리 기능의 종류

자체 모델러를 가진 기존 열전도 해석 사용 프로그램은 Fig. 2와 같이 모델 데이터의 표면을 처리하는 기능을 가지고 있다.

이러한 기능을 본 연구에서는 표면 전처리 기능으로 구분하였으며, Fig. 2에 제시된 기능을 정리하면 다음과 같다.

①의 기능은 삼각형화 기능, ②의 기능은 예각의 폴리라인을 둔각으로 변환 기능, ③의 기능은 예각의 삼각형을 최소화 하는 변환 기능, ④의 기능은 삼각형의 Normal Vector의 일치하게 하는 기능, ⑤의 기능은 폴리라인의 간섭을 없애는 변환 기능, ⑥의 기능은 폴리곤의 간섭을 배제는 변환 기능, ⑦의 기능은 매쉬의 홀을 채우는 변환 기능, ⑧의 기능은 매 접면의 정점을 매칭해주는 변환 기능, ⑨의 기능은 여러 엣지(edge)중 엣지를 정리 하는 기능 등이 있다. 본 연구에서는 STL모델 파일을 개발 프로그램에서 해석모델로 이용할 때 오류를 보완하고 열전도 해석을 위한 mesh 생성이 가능하도록 import된 모델의 표면처리에 필요한 4가지 기능을 연구 개발하였다.

2.2. 정점 오류 보정 필터

정점 오류 보정 필터는 동일 좌표 위치하여야 하는 두개의 정점을 표현하는 데 사용자의 미세한 오류로 모델이 작성되거나 Fig. 3과 같은 STL파일포맷³⁾의 특성상 실수 연산에 따른 연산오류로 두 정점의 좌표가 미세하게 다르게 되는 오류가 발생한다. 이러한 오류를 보정하여 열전도 해석을 위한 mesh 생성을 원활히 하기 위한 필터이다.

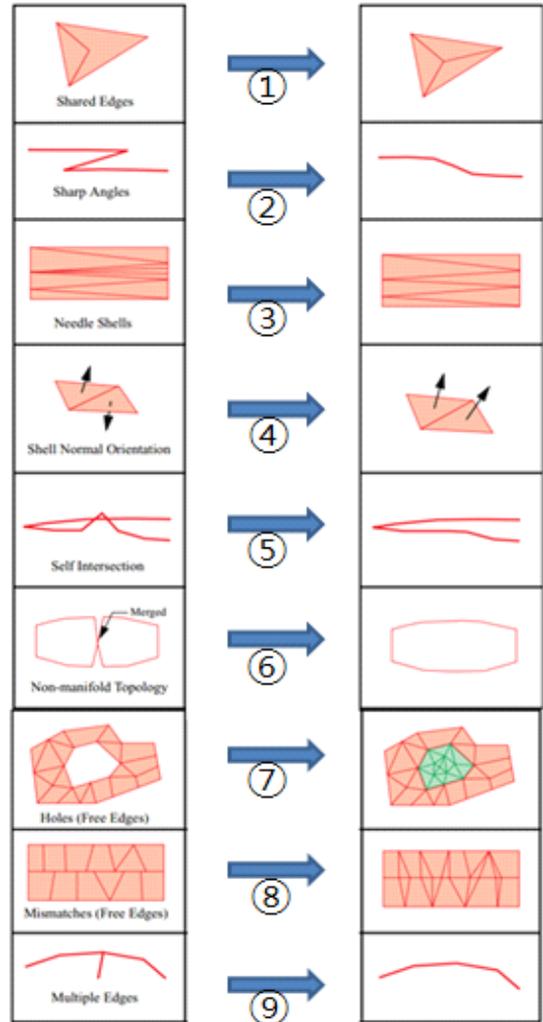
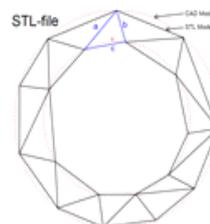


Fig. 2. methods for fixing surface Mesh and Vertex



```

ASCII STL
solid name
facet normal ni nj nk
  outer loop
    vertex v1x v1y v1z
    vertex v2x v2y v2z
    vertex v3x v3y v3z
  endloop
endfacet
    
```

Fig. 3. format of STL file



Fig. 4. process of vertex error fixing

STL파일은 Solid 모델이며 모든 엣지는 두면에 접한다는 특징을 이용하여 Fig. 4와 같이 오류 엣지를 추출하고 정점과 정점 사이의 문턱치(threshold)를 확인하여 문턱치 이하인 경우 동일 정점으로 보정하고 모델을 재구성한다.

pseudocode로 생성된 Algorithm1은 STL파일을 읽고 정점, 엣지, 삼각면 등의 자료구조로 구성하고 엣지의 중복 유무를 비교하여 오류 정점이 포함된 엣지를 추출하는 알고리즘이다.

그리고 Algorithm2는 Algorithm1에서 구성된 자료구조와 추출된 오류 엣지를 가지고 문턱치를 이용한 보정된 결과, 삼각형의 두점 이상을 동일한 경우 자료구조에서 제외시켜서 매쉬를 재구성하여 보정된 STL파일을 구성한다.

해당 기능의 구현을 위한 기초 coding은 다음과 같다.

Algorithm1

function 오류 엣지의 추출

1. STL file 로딩
2. 삼각 mesh 정보로 vertexarr array(x,y,z), edgearr array (vertex Index1,vertex Index2, face index1,face index2), facearr array(vertex Index1,vertex Index2,vertex Index3) 생성
3. for edge1 in edgearr array
4. flag = abnormal state
5. for edge2 in edgearr array
6. if edge1 = edge2 then
7. Continue
8. end if
9. if edge1.start point = edge2.end point and edge1.end point = edge2.start point then
10. flag = Normal state
11. end for
12. end for

Algorithm2

function 모델의 재구성

1. newfacearr array(vertex Index) 생성
2. for face in facearr array(기존 삼각매쉬)
3. if face.vertex1 = face.vertex2 then Continue
4. end if
5. if face.vertex2 = face.vertex3 then Continue
6. end if
7. if face.vertex3 = face.vertex1 then Continue
8. end if

9. newfacearr.add(face)
10. end for
11. facearr =newfacearr

2.3. Normal vector 오류 보정 필터

Normal vector의 오류 보정 필터는 다양한 모델 저작툴에서 생성되는 STL파일의 Normal vector 오류를 최소화하기 위한 필터이다.

열전달 해석에 사용되는 모델은 Solid 모델이며 모델을 이루는 각 삼각형의 Normal vector가 모델의 외부를 향하고 있고 모든 엣지는 두면에 접한다는 특징을 가지며, 이와 같지 않은 벡터의 오류를 찾아서 수정을 한다.

첫 번째, 형상 모델 표면의 각 삼각형의 Normal vector가 모델의 외부를 향하는 지 알아내는 문제가 있는 데 불록다면체일 경우는 중심점을 이용하여 쉽게 처리가 가능하나 오목다면체나 홀이 있는 형태인 경우 내외의 구분이 어렵다. 그래서 오목다면체를 불록다면체와 나누어 처리하기도 한다.

본 연구에서는 2차원의 폴리곤 인 사이드 검사 방법¹¹⁾을 3D로 확장¹⁰⁾하여 Fig. 5과 같이 객체의 폴리곤들과 직선의 간섭으로 안쪽과 바깥쪽을 구분한다.

Algorithm3은 Algorithm1에서 구성된 자료구조를 바탕으로 모든 삼각면의 Normal vector를 구하고 Normal vector를 라인화하고 모델과의 간섭으로 안쪽과 바깥쪽으로 구분한다. 그러나 이러한 방법은 시간의 복잡도 $O(n^2)$ 로 연산 시간이 삼각면의 수에 따라 기하급수적으로 증가한다. 이를 피하기 위해 Algorithm4와 같이 시작이 되는 삼각면의 벡터를 안쪽과 바깥쪽으로 구분하고 이것을 기준으로 Fig. 6과 같이 삼각면을 이루는 정점의 순서의 방향과 접한 삼각면은 인접한 edge부분에서 반대방향을 가지는 특징을 재귀함수화하여 인접 삼각면으로 변져나가는 형태로 개발 되어 시간의 복잡도가 $O(n+\log(N))$ 로 연산시간의 개선하였다.

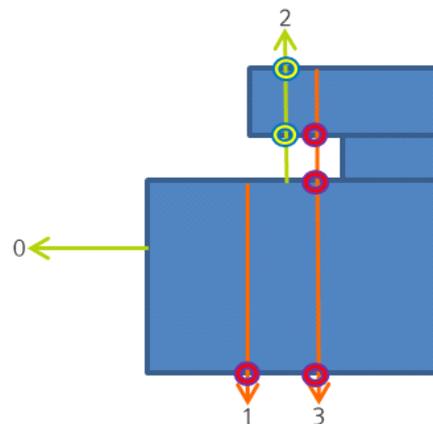


Fig. 5 concept of checking interference between vector and polygon

해당 기능의 구현을 위한 기초 coding은 다음과 같다.

Algorithm3

function Normal vector

1. for face1 in facearr array(기존 삼각매쉬)
2. Normalvector1 = getNormalvector(face1)
3. CentorPoint1 = getFaceCentorPoint(face1)
4. line1=setLine(CentorPoint1,CentorPoint1+Normalvecto
r1*10000)
5. Intersectioncount=0
6. for face2 in facearr array(기존 삼각매쉬)
7. if face1 = face2 then
8. Continue
9. end if
10. if checkLineFaceIntersection(line1,face2)=True then
11. Intersectioncount =Intersectioncount+1
12. end if
13. end for
14. if ModulusOperator(Intersectioncount,2) = 1 then
15. error_normal_vectorfacearr.add(face1)
16. end if
17. end for

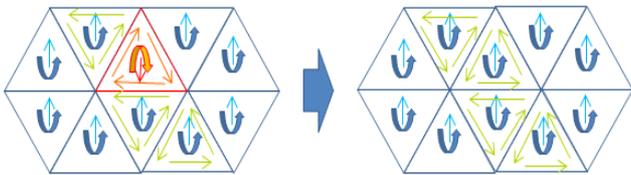


Fig. 6 fixing Normal vector according to the order of vertex in triangle surface

Algorithm4

function Normal vector

1. face1 =first face in facearr array(기존 삼각매쉬)
2. Normalvector1 = getNormalvector(face1)
3. CentorPoint1 = getFaceCentorPoint(face1)
4. line1=setLine(CentorPoint1,CentorPoint1+Normalvector1*
10000)
5. Intersectioncount=0
6. for face2 in facearr array(기존 삼각매쉬)
7. if face1 = face2 then
8. Continue
9. end if
10. if checkLineFaceIntersection(line1,face2)=True then
11. Intersectioncount =Intersectioncount+1
12. end if
14. if ModulusOperator(Intersectioncount,2) = 1 then
15. basicNormalvector= Normalvector1
16. else.
17. basicNormalvector= Normalvector1 * (-1)
18. end if
19. recursive_funtion(face1)

recursive_funtion(face1)

1. if face1.recursive_flag =false then
2. return
3. end if
4. face1.recursive_flag =true
5. if CheckVertexOrderDirection(face1, face1.nearface1) =
same then
6. reverseVertexOrderDirection(face1.nearface1)
7. recursive_funtion(face1.nearface1)
8. else
9. recursive_funtion(face1.nearface1)
10. endif
11. if CheckVertexOrderDirection(face1, face1.nearface1) =
same then
12. reverseVertexOrderDirection(face1.nearface1)
13. recursive_funtion(face1.nearface1)
14. else
15. recursive_funtion(face1.nearface1)
16. endif
17. if CheckVertexOrderDirection(face1, face1.nearface1) =
same then
18. reverseVertexOrderDirection(face1.nearface1)
19. recursive_funtion(face1.nearface1)
20. else
21. recursive_funtion(face1.nearface1)
22. endif

2.4. 접면 추출 필터

접면 추출 필터는 복합재질의 STL파일을 import했을 때 이질 재료가 접하는 접합면을 추출하여 접합면에 대하여 해석을 위한 별도의 mesh를 생성할 수 있도록 하기 위한 사전 작업으로 Fig. 7과 같이 접하는 면을 포함하는 두 모델들의 접면을 삼각매쉬로 세분화하고 접면을 추출하여 이후 매쉬어에서 매쉬화 처리가 원활하게 한다.

접면 추출은 CSG(Constructive solid geometry)의 Boolean Operation1)을 적용한다. Boolean Operation은 VTK⁹⁾의 CSG 라이브러리를 적용하여 두 모델간의 교차면(Intersection)을 통하여 접면을 구하고 빼냄 기능(Subtraction)을 통하여 접면과 객체의 접면간에 정점이 일치하도록 모델의 접하는 면에 정점이 추가 되고 접면의 라인을 따라 삼각화처리가 되어진 객체를 생성한다.

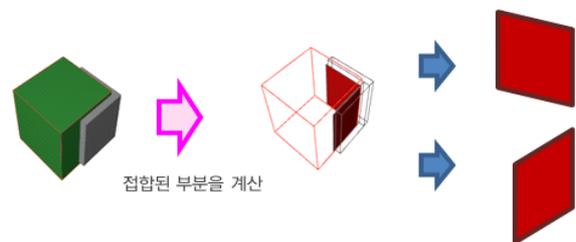


Fig. 7. process of extracting intersection surface

2.5. 폴리곤 라인 추출

폴리곤 라인 추출은 Fig. 8과 같이 객체 이루는 인접한 두 삼각면의 Normal vector의 방향이 다를 때 두 삼각면의 접한 모서리를 추출하여 폴리곤 라인을 생성하는 것으로 후처리 과정(Post Process)에서 솔버(solver)를 통한 해석 결과를 그래픽적으로도 출력할 때 칼라 맵(color map)이 적용된 모델에서 객체의 윤곽을 표현해주는 라인이 되고 STL파일을 매쉬어(mesher)에서 접면을 처리하기 위한 사전 작업이다. Algorithm1에서 구성된 자료 구조를 바탕으로 모든 삼각면의 Normal vector를 구하여 인접 삼각면과 비교하고 다를 경우 엣지를 추출하고 추출된 엣지를 폴리곤 라인으로 재구성하고 매쉬어와 Post process에서 사용가능한 파일포맷으로 출력한다.

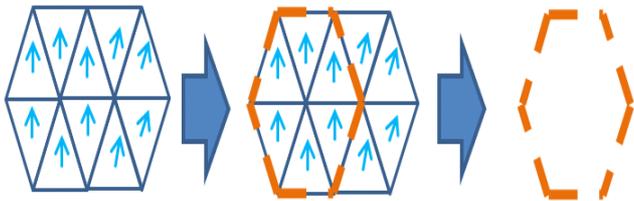


Fig. 8. Process of extracting external poly-line

3. 개발 기능의 프로그램 구현 결과

3.1. 열전도 해석을 위한 표면 전처리 기능의 구현

정점 오류 보정필터는 Fig. 9와 같이 틈이 생긴 면을 가진 모델에서 보정 후 틈이 매워진 모델로 재구성되어진 보정 결과를 보여주고 있다.

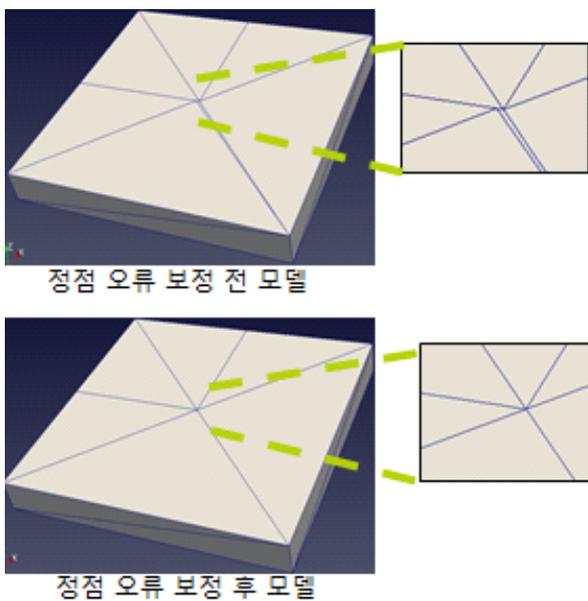


Fig. 9. example of fixing vertex error

Normal vector 오류 보정 필터는 Fig. 10과 같이 청색은 외부로 Normal vector를 가진 삼각면과 적색은 내부로 Normal vector를 가진 삼각면 되어진 모델을 Normal vector의 오류 보정하여 청색은 외부로 Normal vector를 가진 삼각면으로만 이루어진 모델로 재구성된 결과를 보여준다.

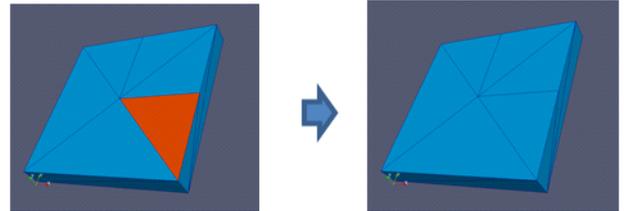


Fig. 10. example of normal vector error(blue surface's normal vector heads toward exterior, red's interior)

접면 추출 필터는 Fig. 11과 같이 두 모델과의 Intersection Operation을 통하여 접면을 구하고 두 모델과의 Subtraction Operation을 통하여 접면과 객체 접면의 정점이 일치하도록 모델의 접하는 면에 정점이 추가되고 접면의 라인을 따라 삼각화처리가 되어진 모델로 재구성된 결과를 보여주고 있다.

폴리곤 라인 추출은 Fig. 12과 같이 STL모델에서 폴리곤 라인 추출되어진 파일을 생성한 결과를 보여준다.

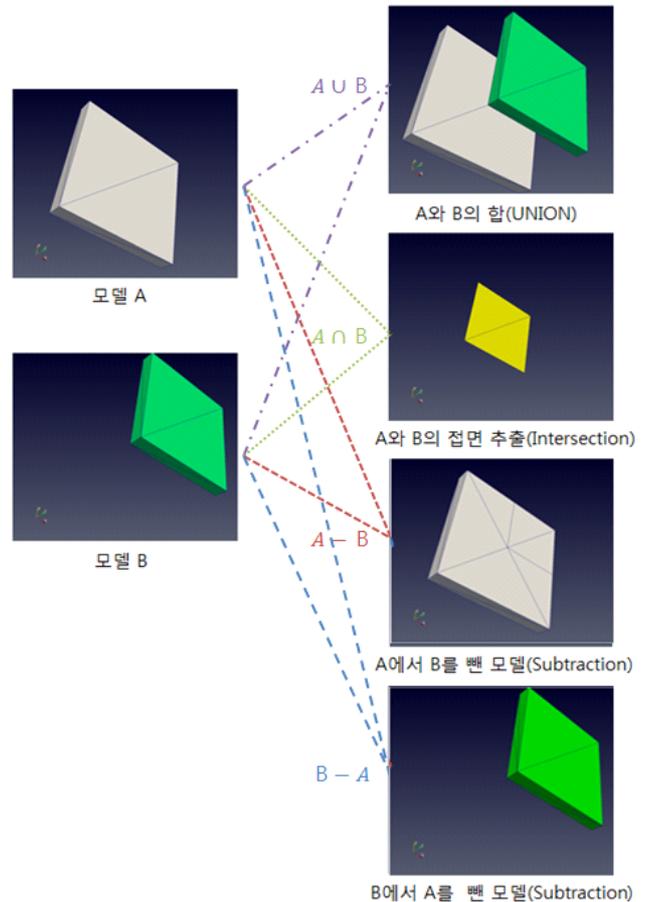


Fig. 11. example of extracting intersection surface by boolean operation

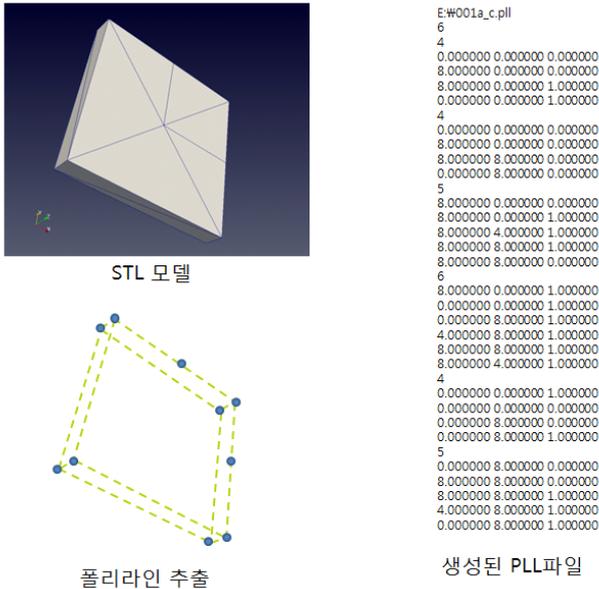


Fig. 12. example of extracting poly-lines

4. 결론

본 연구에서 개발을 진행하고 있는 3차원 정상상태 전열해석 프로그램의 전처리과정(pre-process)에서 요구되는 주요 기능을 개발하여 이를 개발 프로그램에 구현하였으며, 개발되는 프로그램에 요구되는 기능의 적용이 가능할 것으로 사료된다.

본 연구에서 구현한 기능을 간략히 정리하면 다음과 같다.

첫째, 전열해석을 위한 3차원 모델의 형상 데이터에 대하여 각 정점의 오류를 추출하고 수정하는 기능(정점 오류 보정 필터)을 개발하였으며, 이에 따라 개발 프로그램에 import 되는 3차원 모델의 형상 데이터 오류를 용이하게 수정하고 해석을 위한 mesh 생성이 원활하기 이루어 질 수 있을 것으로 판단된다.

둘째, 하나의 3차원 형상을 구성하는 각 표면의 Normal vector가 지향하는 방향이 다른 경우 발생될 수 있는 오류를 추출하고 수정할 수 있는 기능(Normal Vector 오류 보정 필터)을 개발하였으며, Normal vector의 불일치로 인한 해석 모델의 mesh 생성시 발생하는 오류를 방지하여 해석 모델의 원활한 생성이 가능할 것으로 판단된다.

셋째, 해석을 위한 모델이 이질 자재로 구성되어 있을 경우 이질 자재간의 접합면을 추출하여 새로운 접합면을 구분 및 생성할 수 있는 기능(접면 추출 필터)을 개발하였다. 이에 따라 이질 재료간의 접합면에 대한 별도의 mesh 생성이 가능할 것으로 판단되며, 이질 재료간 접합면에 추가적인 경계조건의 설정이 가능할 것으로 판단된다.

넷째, 하나의 재질을 가진 3차원 형상의 외곽선만 추출할 수 있는 기능(폴리곤 라인 추출)을 개발하여 개발 프로그램의 전처리(pre-process) 및 후처리(post-process)시 보다 명확한 모델의 형상을 파악할 수 있는 결과 그래픽의 도출이 가능할 것으로 판단된다.

현재 개발된 기능은 비교적 단순한 형상에 대하여 기능 시험이 실시되어 기본적인 기능 구현이 가능한 알고리즘 및 기본 coding은 개발되었으며, 지속적으로 보다 복잡한 형상을 대상으로 기능을 적용하여 문제점을 도출 수정하고 있다.

향후 개발되는 국산 3차원 전열해석 프로그램에 탑재될 기본 기능으로서 지속적으로 기능 향상 및 오류 수정을 진행할 계획이다.

Acknowledgements

본 연구는 국토교통부 주거환경연구사업의 연구비지원(15RERP-B082204-02)에 의해 수행 되었습니다

References

- [1] van Rossen, Sander; Baranowski, Matthew (2011), "Real-time constructive solid geometry", in Ansari, Marwan, Game Development Tools, CRC Press, pp. 79–96, ISBN 9781439867723.
- [2] Glassner, Andrew S. (1989), An Introduction to Ray Tracing, Morgan Kaufmann, p. 80, ISBN 9780122861604.
- [3] The StL Format, Standard Data Format for Fabbers StereoLithography Interface Specification, 3D Systems, Inc., October 1989
- [4] USER GUIDE, pro-STAR WITH AUTO MESH GENERATION VERSION 3.26 © 2005 CD-adapco
- [5] Physibel, <http://www.physibel.be/index.htm>
- [6] VTK, <http://http://www.vtk.org/>
- [7] HEAT, <http://www.buildingphysics.com/index.htm>
- [8] OpenFOAM, <http://openfoam.org/>
- [9] ParaView <http://www.paraview.org/>
- [10] P. C. P. Carvaiho, P. R. Cavalcanti, Point in Polyhedron Testing Using Spherical Polygons, Graphics Gems V edited by A. W. Paeth, Academic Press 1995
- [11] J. O'Rourke, Computational Geometry in C, Cambridge, Second Edition, Academic Press 1998