

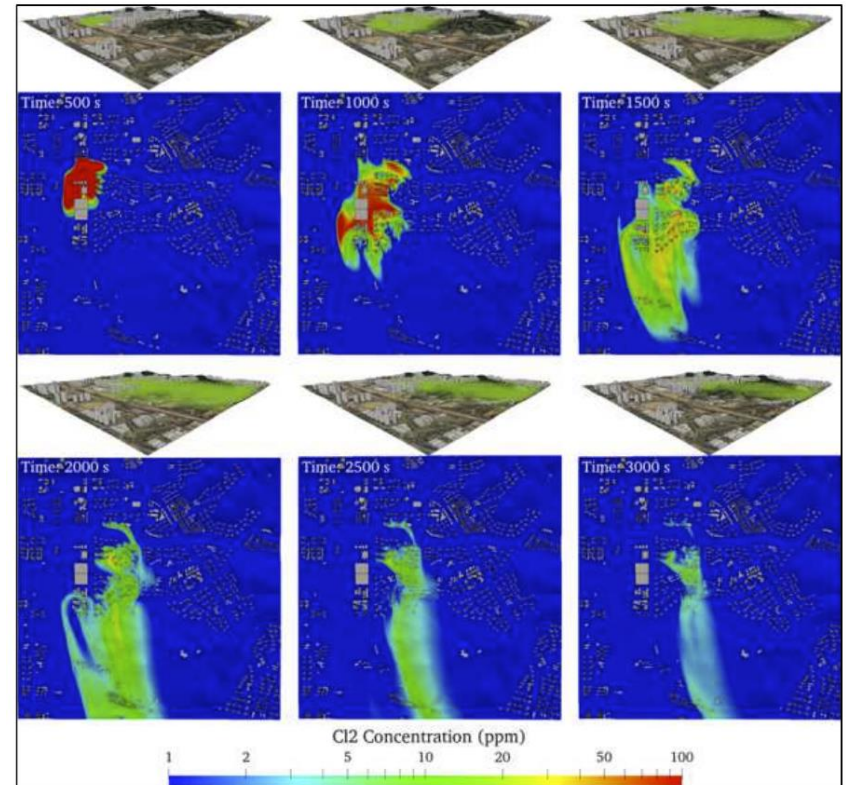
오픈소스를 활용한
화학물질누출사고 영향 범위 해석



김현식, 길재흥,
박현강, 유지형

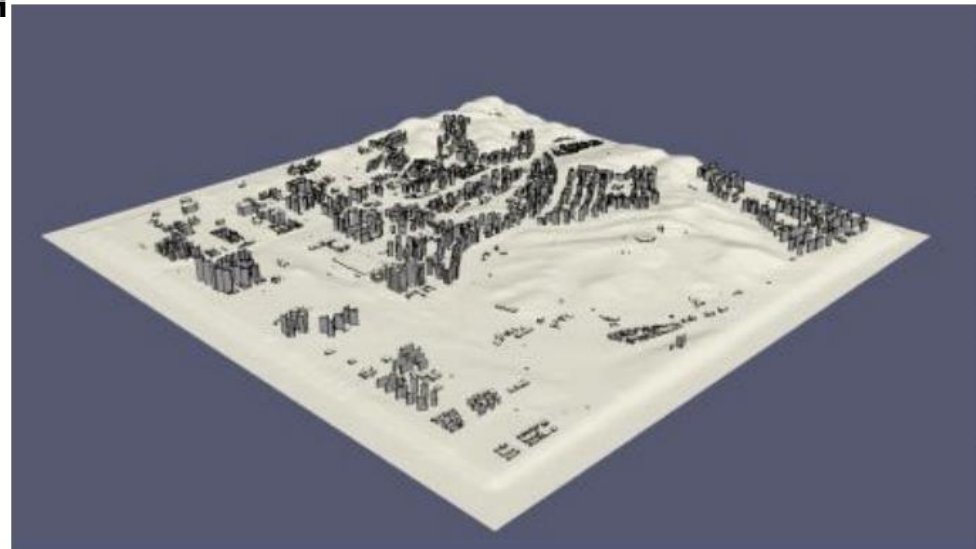
1. 연구 목표 및 내용
2. 해석
 1. 해석 개요
 2. 해석 시나리오
 3. 전처리
 4. 해석 조건
 5. 해석 결과
3. 결론 및 후기

- 화학물질 누출사고는 산업 현장뿐만 아니라 일상생활에서도 발생할 수 있는 재난으로 화학물질 누출 사고가 일어날 경우 그 확산 범위를 예측하는 것이 사고 대응에 있어 매우 중요하며
- 최근 화학물질 누출사고 영향 범위 해석은 다양한 사고 요인들을 고려하는 기술 개발이 수반되어야 한다
- 따라서 본 연구에서는, 기능 확장이 용이한 오픈소스 CFD 해석 프로그램인 OpenFOAM을 사용하여 기존의 현상학적모델(SLAB), 기상데이터 그리고 3D 지도데이터 등을 활용하여 화학물질 누출사고 영향 범위를 예측하고자 하였다



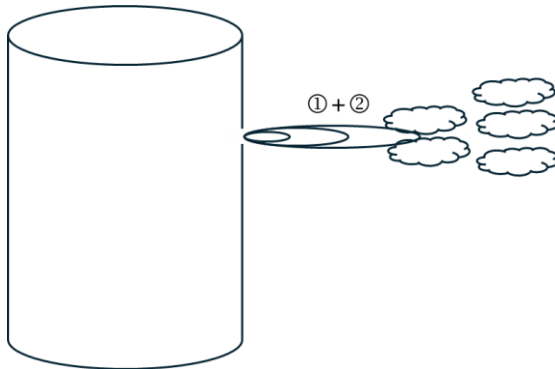
- OpenFOAM 시나리오 해석 개요

- 지형 및 건물 형상 생성
 - Python 패키지를 이용한 DEM shp 파일 데이터 추출
 - ParaView 필터와 OpenCASCADE를 이용한 형상 STL 파일 생성
- blockMesh와 snappyHexMesh를 이용한 격자생성
 - SLAB 모델을 활용한 해석 영역 선정
- 해석 시나리오에 맞는 경계조건 설정
- 지형풍 초기화를 위한 정상상태 해석
- 화학물질 Source term 설정
- 화학물질 누출 Transient 해석
- ParaView를 이용한 결과 해석

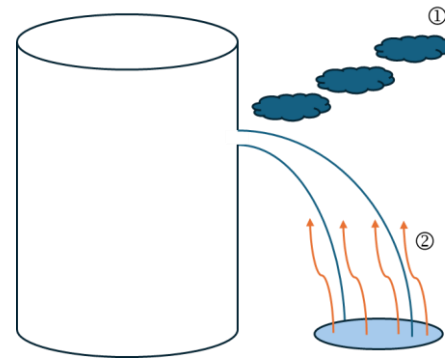


- 해석 지형: 경부고속도로 동탄 터널 인근 (4km X 4km)
- 기상 조건
 - 대기 안정도: Pasquill class D
 - 풍속: 측정지점 해발도 10m에서 풍속 2.41m
 - 풍향: 270° (→, 서풍)
 - 대기 온도: 25°C
- 화학물질 누출 시나리오
 - 누출 지점: 경부고속도로 동탄 터널 인근
 - 누출 방향: 남쪽 방향
 - 누출 물질: 암모니아 (NH₃)
 - 총 누출량: 10.83 ton

- Case1
 - 시나리오: 10분 전량누출, 기상 누출
- Case2
 - 시나리오: 10분 전량누출, 2상(기상+액상) 누출
- Case3
 - 시나리오: 누출률 고정, 기상 누출
- Case4
 - 시나리오: 누출률 고정, 2상(기상+액상) 누출



기상 누출



2상(기상+액상) 누출

- 누출률 계산

- 시나리오 상황에서의 밀도, 온도, 잠열등을 바탕으로 계수 N 및 Q 계산

$$N = \frac{\Delta H_V K}{2(P_1 - P_a) \rho_L C_D^2 (\rho_G^{-1} - \rho_L^{-1})^2 T_1 C_{P_L}} + \frac{L_P}{L_e}$$

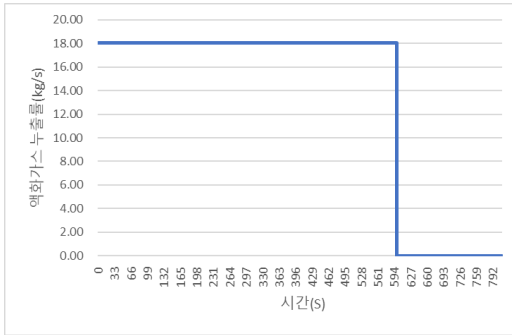
$$Q = \left[\frac{A \Delta H_V}{\rho_G^{-1} - \rho_L^{-1}} \right] \left[\frac{K g_c}{T_1 C_{P_L} N} \right]^{\frac{1}{2}}$$

- 기상 누출일 경우 jet파 분출만 고려, 2상 누출일 경우 jet파 + pool 증발까지 고려
 - Jet 파 분출: 누출공으로 부터 10m 떨어진 지점에서의 속도를 계산
 - Pool 증발: 시간에 따라 변화하는 액체층의 표면적 계산 -> 계산된 표면적으로 증발속도 및 시간당 증발량 계산

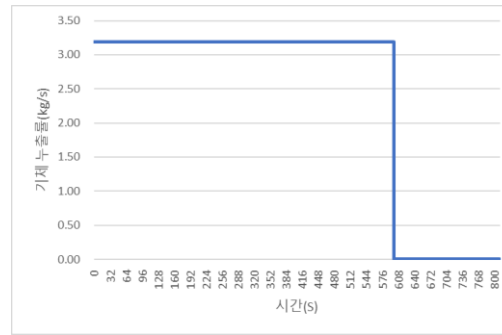
$$A_L = \frac{(Q/\rho_L)}{h_P}$$

$$R_E = \frac{1.4 \times U^{0.78} \times M_W^{2/3} \times A_L \times P_V}{82.05 \times T}$$

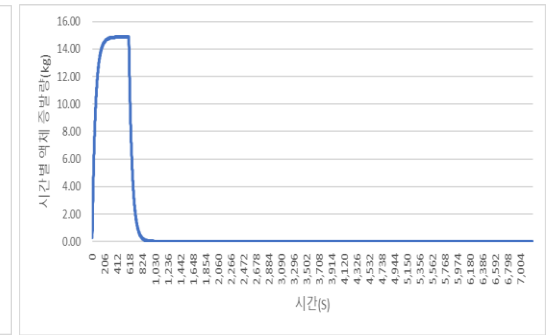
• 시간당 누출량



Jet파 누출
Case1

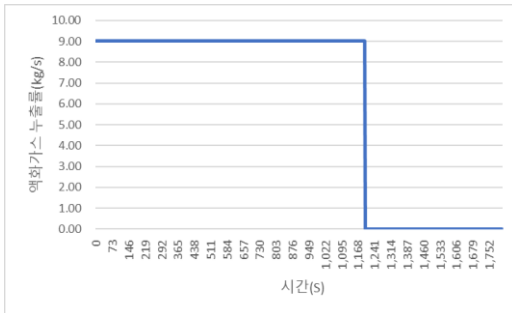


Jet파 누출

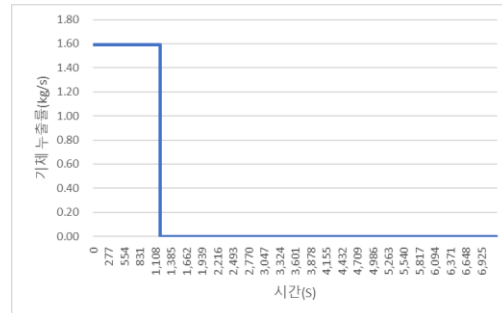


Pool 누출

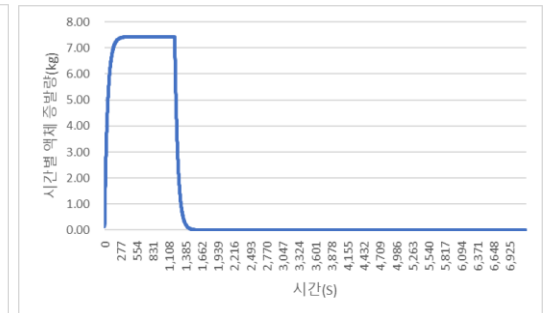
Case2



Jet파 누출
Case3



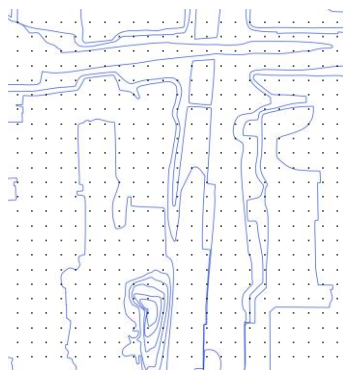
Jet파 누출



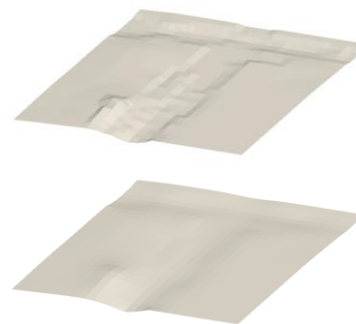
Pool 누출

Case4

- 주요 사용 프로그램(패키지) : ParaView, Python(osgeo, numPy, Scipy)
 - Osgeo 패키지를 이용하여, 해석 영역을 포함하는 수치지형도의 F0010000.shp 파일을 읽어들이
 - 등고선을 이루는 점들의 좌표와 그 높이 정보를 추출
 - 가로, 세로로 일정한 간격을 가지는 grid point를 생성, 선형보간을 통해 grid 점들의 고도를 계산
 - Paraview에서 Grid point끼리 delauny 삼각분할을 통해 지형 표면을 생성
 - Paraview에서 지형 표면을 부드럽게 처리해주는 필터(smooth)를 사용

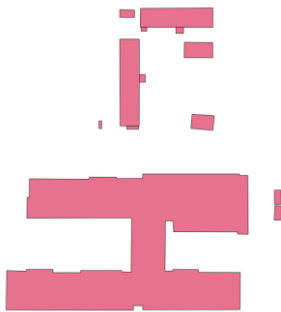


Qgis로 보는 등고선과, Grid point 예시

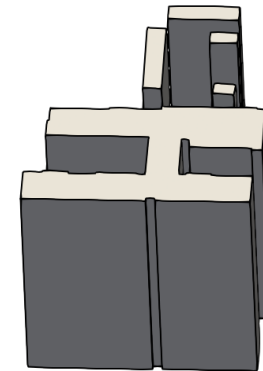


ParaView를 통해 생성/가공한 지표면

- 주요 사용 프로그램(패키지) : Python(osgeo, PythonOCC)
 - Osgeo 패키지를 이용하여, 해석 영역을 포함하는 수치지형도의 B0010000.shp 파일을 읽어들이
 - 건물을 이루는 점들의 좌표와 그 건물의 층수 정보를 추출
 - 건물 평면의 무게중심에서, 이전에 생성한 지형의 고도를 계산하여 층수*3을 더하여 최고 고도 계산 (건물 면적이거나, 층수를 기준으로 생성에서 제외할 수 있음)
 - 해발고도 0m 에서부터 최고 고도까지 돌출(extrusion)기능으로 solid 생성



명칭	구분	종류	용도	주기	층수
1	NULL	주택외건물	근린생활시설	NULL	
2	NULL	주택외건물	기타시설	동탄교역지...	
3	NULL	주택외건물	기타시설	동탄교역지...	
4	NULL	주택외건물	교육연구시설	다원초교	
5	NULL	무벽건물	기타시설	동탄교역지...	
6	NULL	무벽건물	기타시설	NULL	
7	NULL	무벽건물	기타시설	동탄교역지...	
8	NULL	무벽건물	기타시설	동탄교역지...	
9	NULL	무벽건물	기타시설	동탄교역지...	
10	NULL	운실	기타시설	동탄교역지...	
11	NULL	가건물	기타시설	NULL	
12	NULL	가건물	기타시설	동탄교역지...	



Qgis로 보는 건물 평면과 데이터 예시

PythonOCC를 통해 생성한 건물 solid

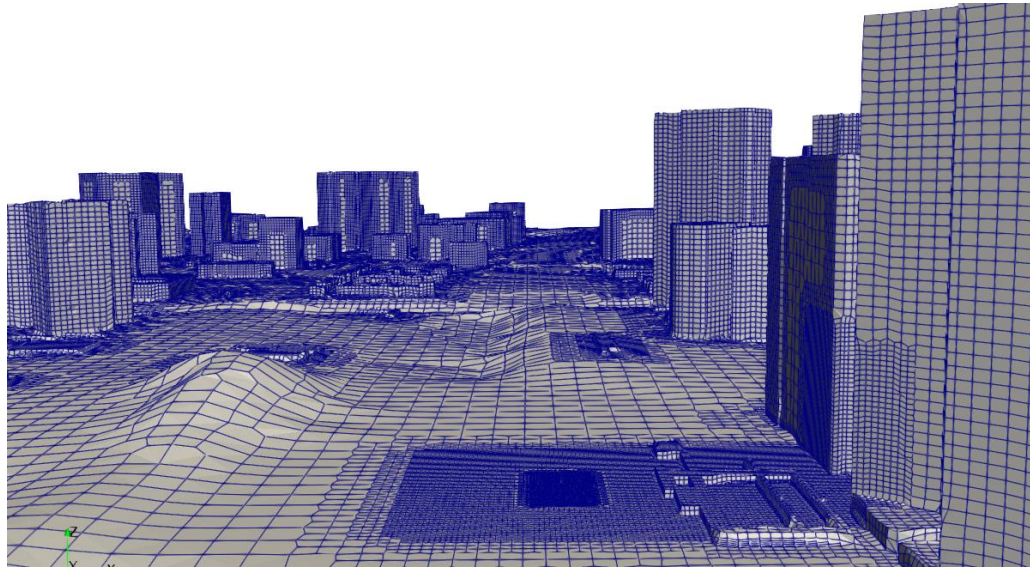
- snappyHexMesh

- STL 형상 파일을 이용해 형상 표면에 맞춰 3차원 격자를 재생성하는 OpenFOAM 유틸리티
- 건물, 지형, 탱크로리, pool에 대해 refinement 및 5개 레이어 추가
- 격자개수: 7,280,753개
- 격자사이즈: Farfield(100m), Terrain(12.5m), building(3~6m), tank(0.2m)

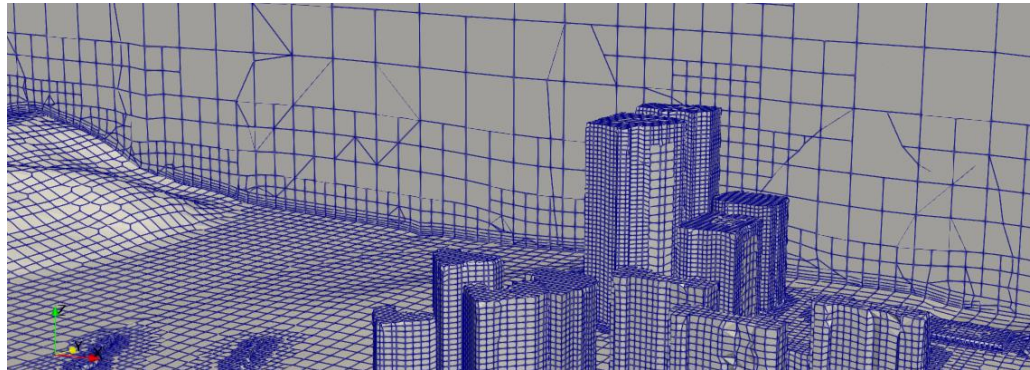
```
105 refinementSurfaces
106 {
107     "building"
108     {
109         level      ( 4 5 );
110     }
111     "terrain1"
112     {
113         level      ( 3 3 );
114     }
115     "terrain3"
116     {
117         level      ( 6 6 );
118     }
119     "pool1"
120     {
121         level      ( 9 9 );
122     }
123     "pool2"
124     {
125         level      ( 9 9 );
126     }
127     "pool3"
128     {
129         level      ( 9 9 );
130     }
131     "tank"
132     {
133         level      ( 9 9 );
134     }
135     zone1
136     {
137         mode        inside;
138         level       (0 0);
139         faceType    baffle;
140         cellZone    movingZone;
141         faceZone    movingZone;
142         cellZoneInside outside;
143     }
```

```
187 addLayersControls
188 {
189     relativeSizes true;
190
191     layers
192     {
193         {
194             nSurfaceLayers 5;
195         }
196     }
197
198     expansionRatio 1.2;
199     firstLayerThickness 0.125;
200     minThickness 0.05;
201     nGrow 0;
202     //featureAngle 50;
203     featureAngle 60;
204     //nRelaxIter 3;
205     nRelaxIter 5;
206     nSmoothSurfaceNormals 1;
207     nSmoothNormals 3;
208     //nSmoothThickness 2;
209     nSmoothThickness 10;
210     maxFaceThicknessRatio 0.5;
211     //maxThicknessToMedialRatio 1;
212     maxThicknessToMedialRatio 0.3;
213     minMedialAxisAngle 90;
214     //nBufferCellsNoExtrude 1;
215     nBufferCellsNoExtrude 0;
216     nLayerIter 50;
217 }
```

- 지형 및 건물 격자



- 경계층 격자



- Pasquill class ATM boundary inlet
 - 지표 열복사와 대기 온도차에 의한 대기 안정도 Pasquill class
 - 입구 경계조건으로 입구 속도 U 와 난류값 k , ϵ 가 고려됨
 - FLACS Technical Report 를 참고해 새로운 OpenFOAM library를 빌드
- 출구조건
 - totalPressure 출구압력 설정
- 지형, 건물
 - No slip 조건 설정

```
dimensions [0 1 -1 0 0 0 0];
internalField uniform (0 0 0);
boundaryField
{
    inlet
    {
        type pasquillAtmBoundaryLayerInletVelocity;
        kappa 0.40;
        Cmu 0.09;
        flowDir (0 0 0);
        zDir (0 0 1);
        Uref 2.41;
        Zref 10.0;
        z0 0.03;
        d 0.0;
        Ls 1.0;
        Zs 0.0;
        value $internalField;
    }
}
```

- OpenFOAM

- OpenFOAM(Open Field Operation and Manipulation)은 C++ 언어로 작성된 연속체 역학을 위한 객체지향 수치해석 라이브러리이며, 1980년대 후반에 영국의 Imperial College에서 최초로 개발
- 공학 및 자연과학 분야의 연구자들이 각자의 분야에서 사용할 수치해석 프로그램을 유연하게 개발할 수 있는 강력한 통합 툴킷을 만들기로 한 것이 그 시작이었으며, C++ 프로그래밍의 객체지향적(Object Oriented) 특징을 최대한 활용하여 제작
- OpenFOAM은 라이브러리이지만 그 라이브러리를 이용하여 제작한 다수의 수치해석 프로그램들도 공개되어 OpenFOAM 패키지에 내장

$$\frac{\partial}{\partial t}(\rho U) + \nabla \cdot \phi U - \nabla \cdot \mu \nabla U = -\nabla p$$

```
solve
(
    fvm::ddt(rho,U)
  +fvm::div(phi,U)
  -fvm::laplacian(mu,U)
  ==
  -fvc::grad(p)
)
```

Open  FOAM

- **Steady 계산 솔버**
 - buoyantSimpleFoam: RANS 또는 LES 난류모델을 사용하고, MRF 기능을 사용할 수 있는 압축성 정상상태 난류유동 해석 프로그램
- **Transient 계산 솔버**
 - buoyantPimpleFoam: RANS 또는 LES 난류모델을 사용하고, 동적격자 기능을 사용할 수 있는 압축성 비정상상태 난류유동 해석 프로그램
- **Turbulenc Model**
 - Realizable k-Epsilon two layer model

$$\frac{\partial}{\partial x_i} (k u_i) = \frac{\partial}{\partial x_j} \left(\left(\nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right) + \nu_t S^2 - \epsilon$$

$$\frac{\partial}{\partial x_i} (\epsilon u_i) = \frac{\partial}{\partial x_j} \left(\left(\nu + \frac{\nu_t}{\sigma_\epsilon} \right) \frac{\partial \epsilon}{\partial x_j} \right) + C_1 S \epsilon - C_2 \frac{\epsilon^2}{k + \sqrt{\nu \epsilon}}$$

- Mass Source

- scalarCodedSource

- Source term을 code 형태로 입력하여 Source에 대한 변화를 implicit하게 계산
 - 해석 시나리오를 기반으로 Jet Source 및 Pool Source 계산

```
39 codeInclude
40 #{
41 #};
42
43 codeCorrect
44 #{
45 #};
46 codeAddSupRho //incompressible flow: codeAddSup, compressible flow: codeAddSupRho
47 #{
48     scalarField& source = eqn.source();
49     const Time& time = mesh().time();
50     const labelList& cellIDs = cells();
51
52     forAll(cellIDs, i)
53     {
54         label cellI = cellIDs[i];
55         source[cellI] -= 3.18533137594124;
56     }
57 #};
58
59 codeConstrain
60 #{
61 #};
```

```
142 scalar AL = accl * timeStep / rhoL / minThick;
143 scalar RE = AL * 1.4 * pow(Uref, 0.78) * pow(Mu, (2.0/3.0)) * Pv / (82.05 * T0);
144 scalar resultL = RE * timeStep / 60;
145
146 totL += resultL;
147
148 Info << "mass flow rate is " << resultL << endl;
149
150 forAll(cellIDs, i)
151 {
152     label cellI = cellIDs[i];
153     source[cellI] -= resultL/nCells;
154 }
155 sleep(1);
156
157 FILE *outFile = fopen("totL_s.txt", "w");
158 fprintf(outFile, "%f\n", totL);
159 fclose(outFile);
160 FILE *outFile2 = fopen("resultL_s.txt", "w");
161 fprintf(outFile2, "%.15f\n", resultL);
162 fclose(outFile2);
163 FILE *outFile4 = fopen("saveTime_s.txt", "w");
164 fprintf(outFile4, "%f\n", curTime);
165 fclose(outFile4);
166
167 }
168 else{
169     scalar resultL;
170
171     FILE *inFile2 = fopen("resultL_s.txt", "r");
172     if (inFile2 != nullptr){
173         fscanf(inFile2, "%Lf", &resultL);
174         fclose(inFile2);
175     }
176
177     Info << "mass flow rate is " << resultL << endl;
178
179     forAll(cellIDs, i)
180     {
181         label cellI = cellIDs[i];
182         source[cellI] -= resultL/nCells;
183     }
184 }
```


- Momentum Source

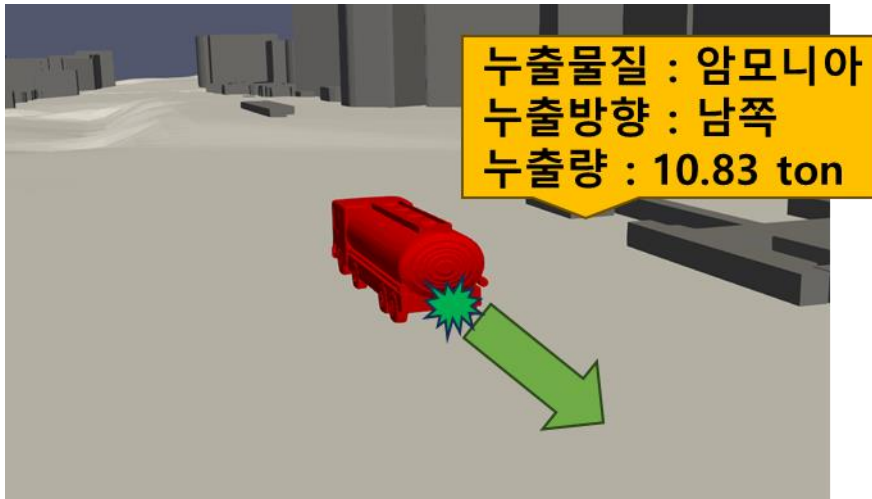
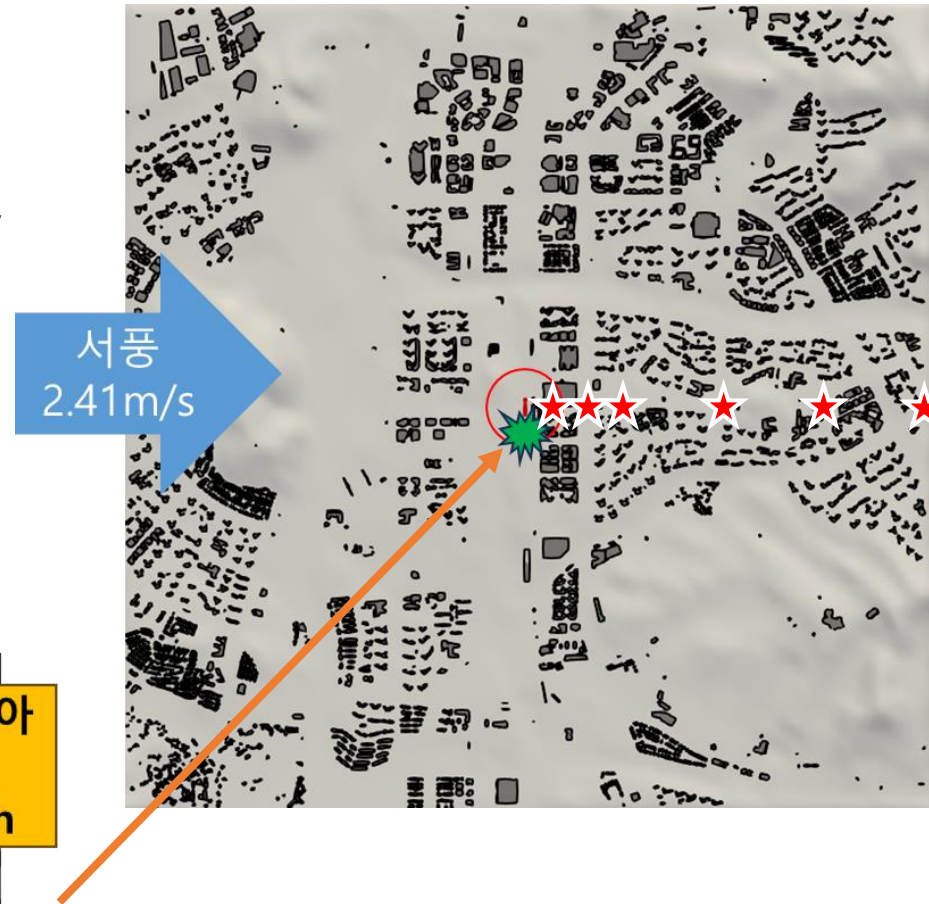
- vectorFixedFunctionConstraint

- 시간에 따른 속도의 함수를 expression 또는 code 형태로 입력할 수 있는 library 개발
 - 해석 시나리오를 기반으로 Jet 및 Pool 에서의 속도 고정

```
91 template<class Type>
92 void Foam::fv::FixedFunctionConstraint<Type>::constrain
93 (
94     fvMatrix<Type>& eqn,
95     const label fieldi
96 )
97 {
98     const Time& time = mesh().time();
99     scalar t = time.value();
100
101     scalar Ux(UxEqn_>value(t));
102     scalar Uy(UyEqn_>value(t));
103     scalar Uz(UzEqn_>value(t));
104
105     Info << "FixedFunctionConstraint>> Ux: " << Ux << " , Uy: " << Uy << " , Uz: " << Uz << endl;
106
107     vector velVec(Ux, Uy, Uz);
108
109     eqn.setValues(cells_, velVec);
110 }
```

```
251 UxEqn
252 {
253     type        expression;
254     expression
255     #{
256         0
257     #};
258 }
259 UyEqn
260 {
261     type        expression;
262     expression
263     #{
264         0
265     #};
266 }
267 UzEqn
268 {
269     type        coded;
270     code
271     #{
272         scalar sTime = 7200; //timeStart
273         scalar curTime = time().value();
274         scalar relTime = curTime - sTime + 1;
275         scalar lTime = 599;
276         scalar saveTime = -1;
277         scalar timeStep = time().deltaTValue();
278     }
```

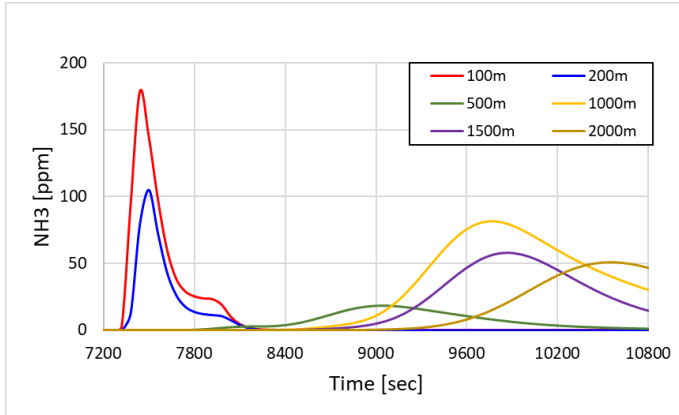
- 암모니아 농도 검출
 - 총 6 곳에서 농도 검출
 - 누출 지점 기준 동쪽으로 100m, 200m, 500m, 1000m, 1500m, 2000m
 - ERPG-1(25ppm), ERPG-2(150ppm), ERPG-3(1500ppm)



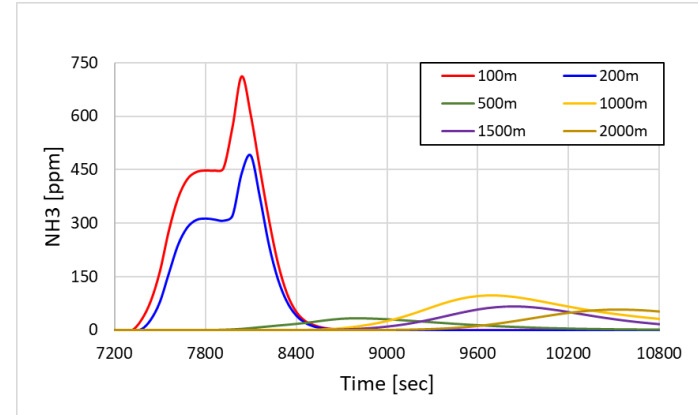
Time-ppm Graph



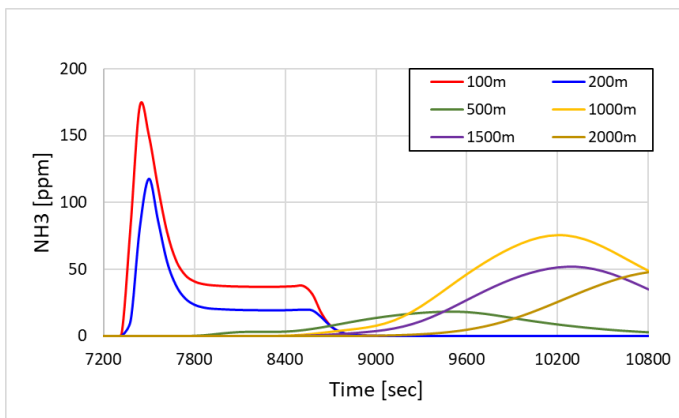
<ul style="list-style-type: none"> Case1 - 10분 전량누출, 기상 누출 	<ul style="list-style-type: none"> Case2 - 10분 전량누출, 2상(기상+액상) 누출
<ul style="list-style-type: none"> Case3 - 누출률 고정, 기상 누출 	<ul style="list-style-type: none"> Case4 - 누출률 고정, 2상(기상+액상) 누출



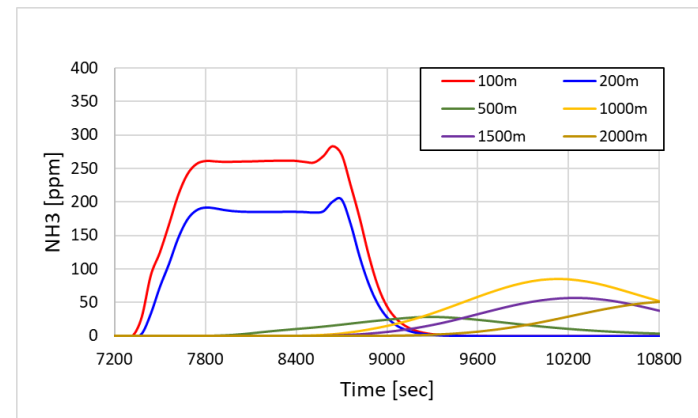
Case1



Case2

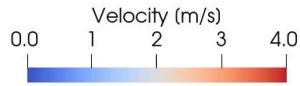


Case3

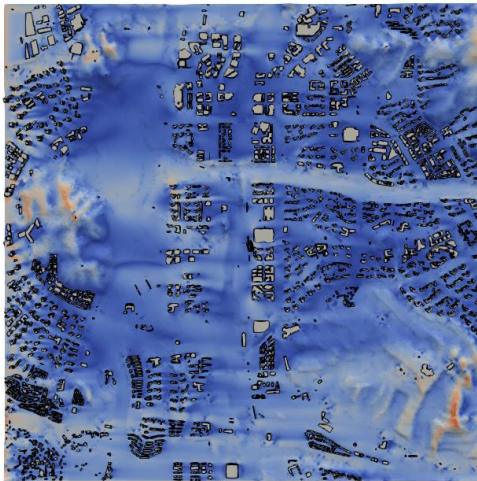


Case4

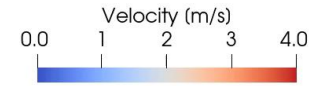
Top view(ERPG1)



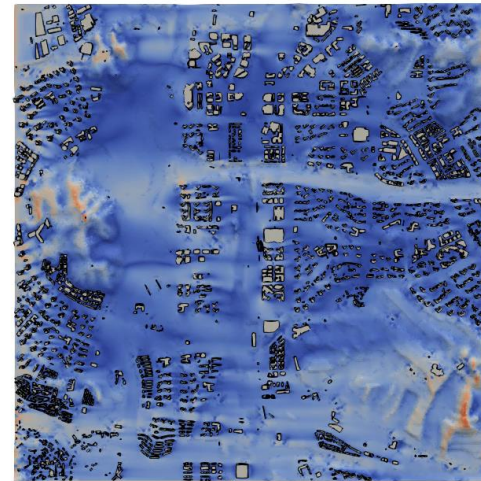
Time: 7200



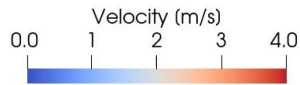
Case1



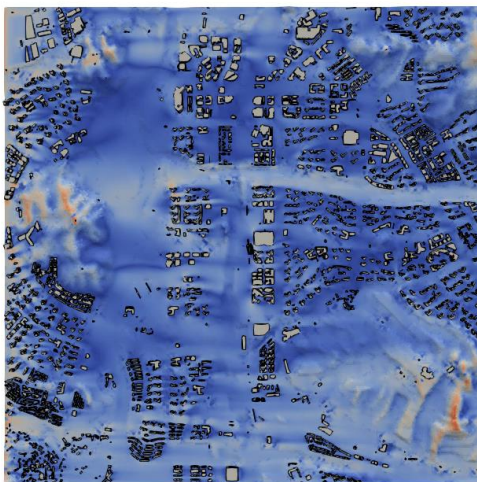
Time: 7200



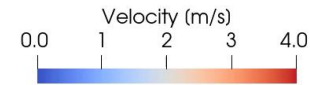
Case2



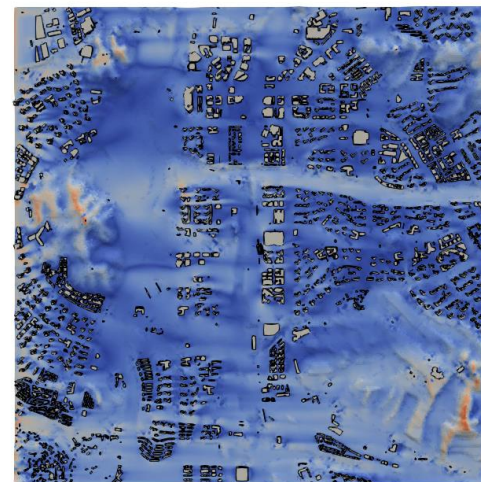
Time: 7200



Case3

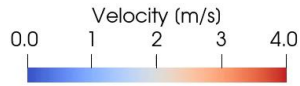


Time: 7200

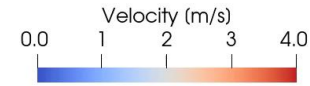
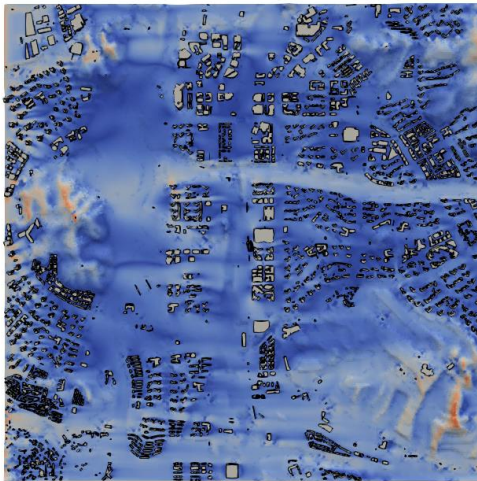


Case4

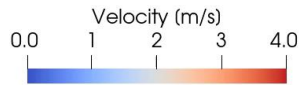
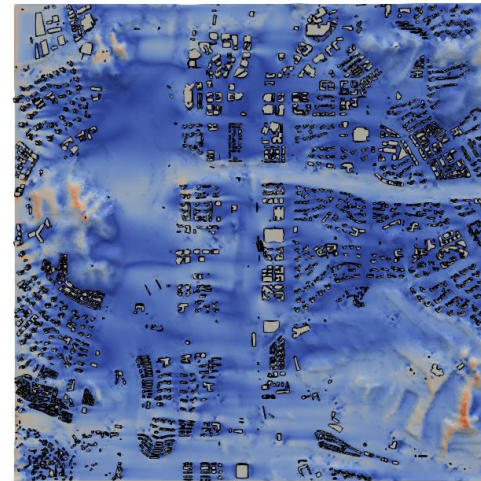
Top view(ERPG2)



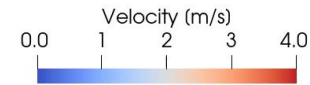
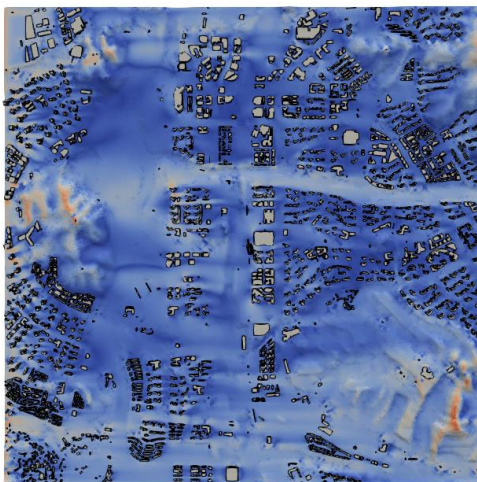
Time: 7200



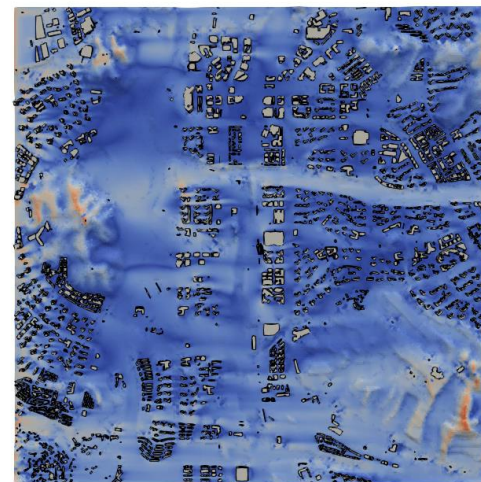
Time: 7200



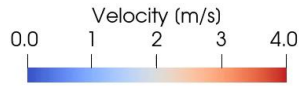
Time: 7200



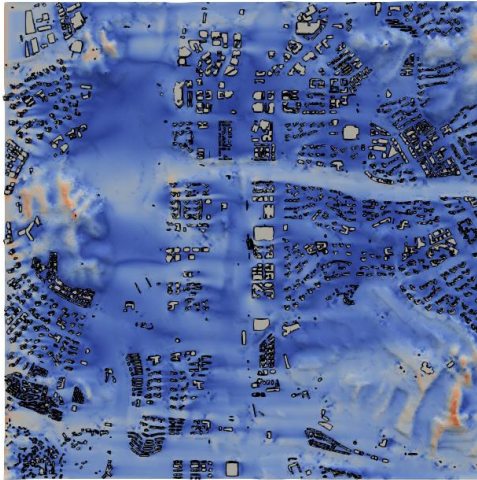
Time: 7200



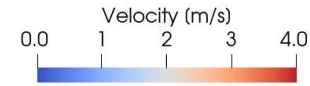
Top view(ERPG3)



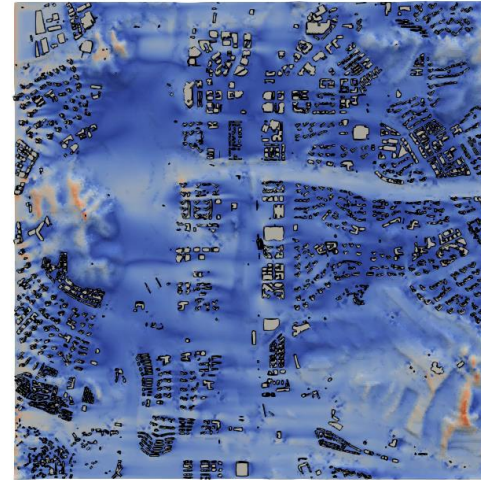
Time: 7200



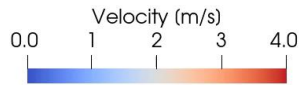
Case1



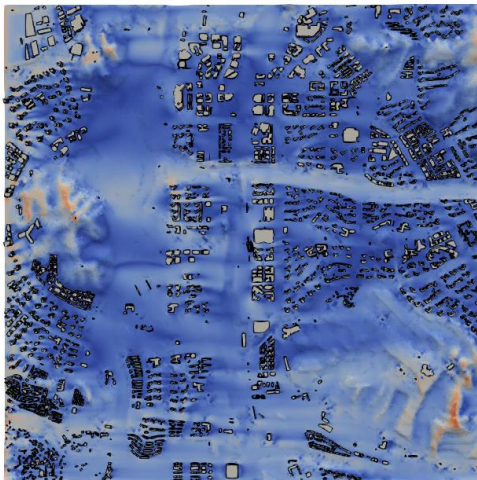
Time: 7200



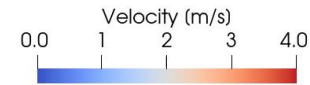
Case2



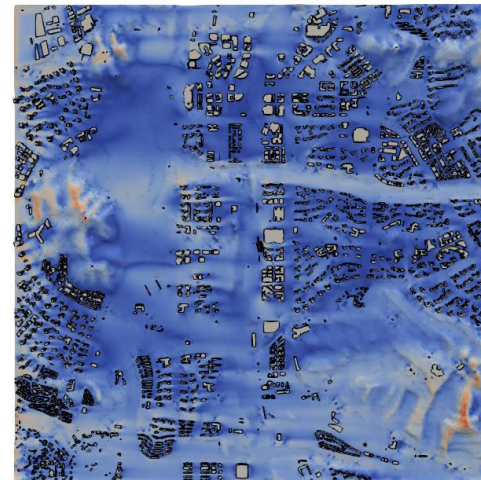
Time: 7200



Case3

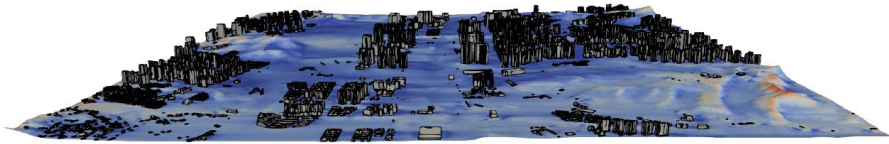
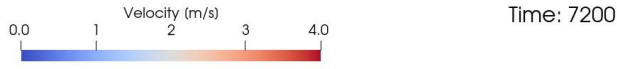


Time: 7200

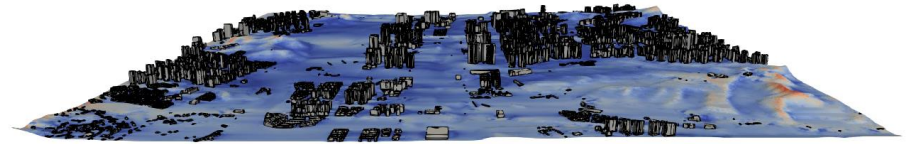


Case4

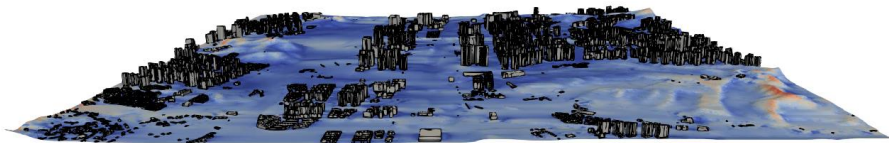
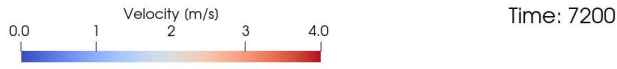
Side view(ERPG1)



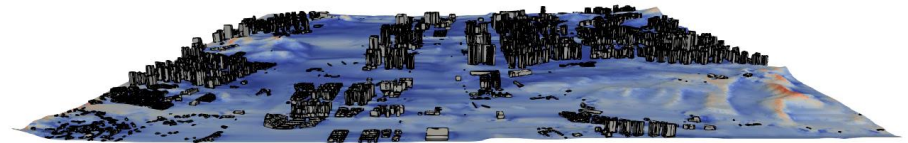
Case1



Case2



Case3

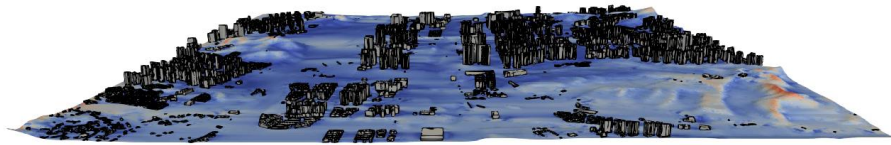


Case4

Side view(ERPG2)



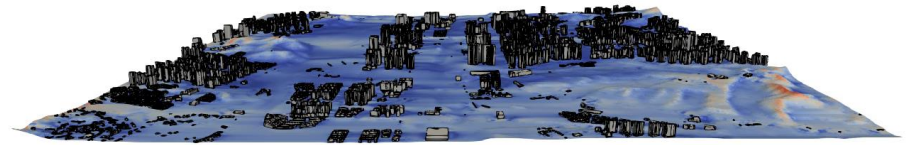
Time: 7200



Case1



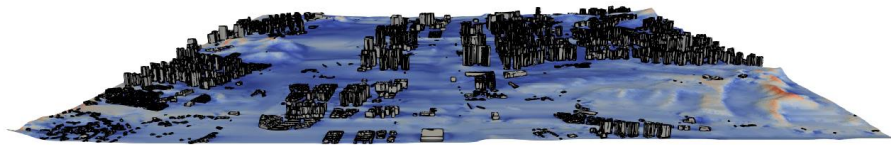
Time: 7200



Case2



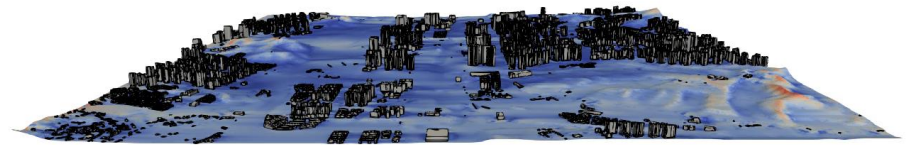
Time: 7200



Case3

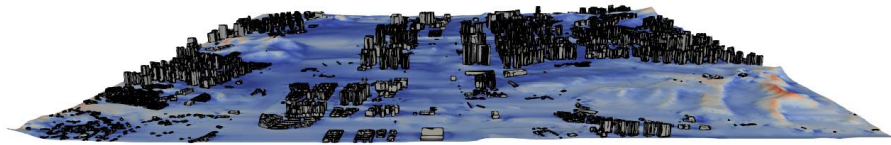
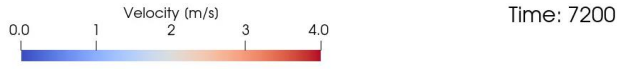


Time: 7200

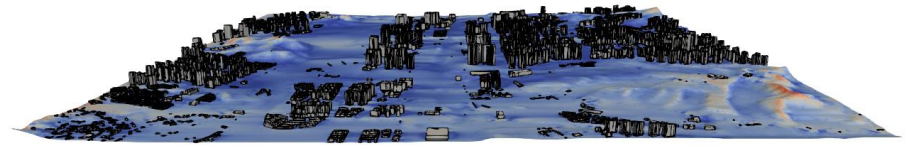


Case4

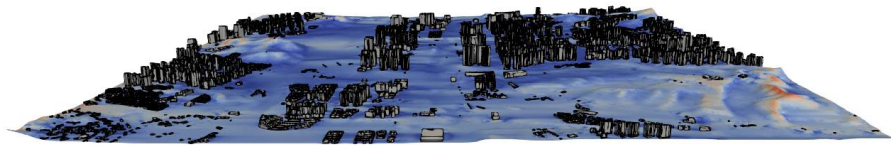
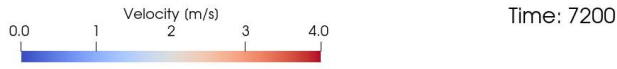
Side view(ERPG3)



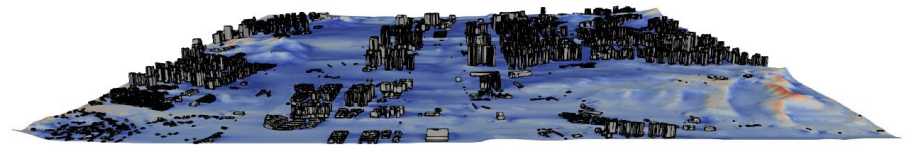
Case1



Case2

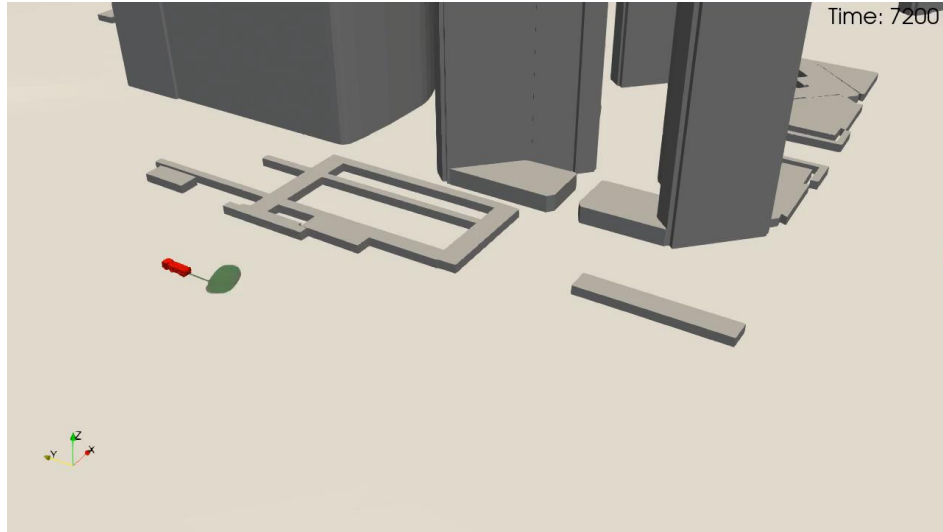


Case3

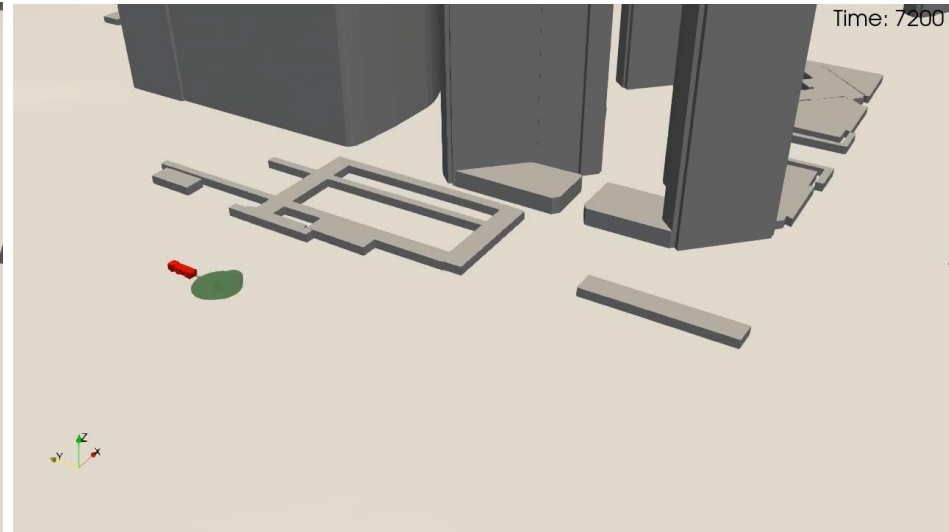


Case4

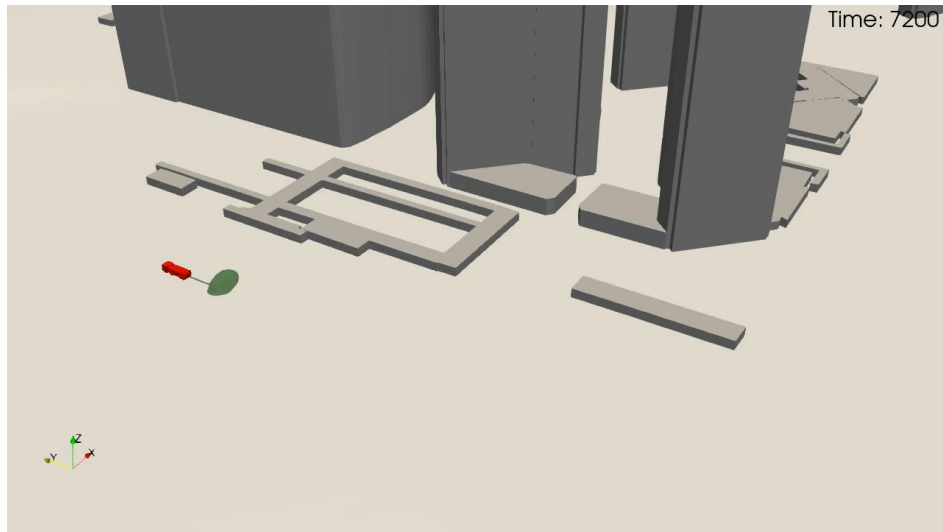
Close-up view(ERPG1)



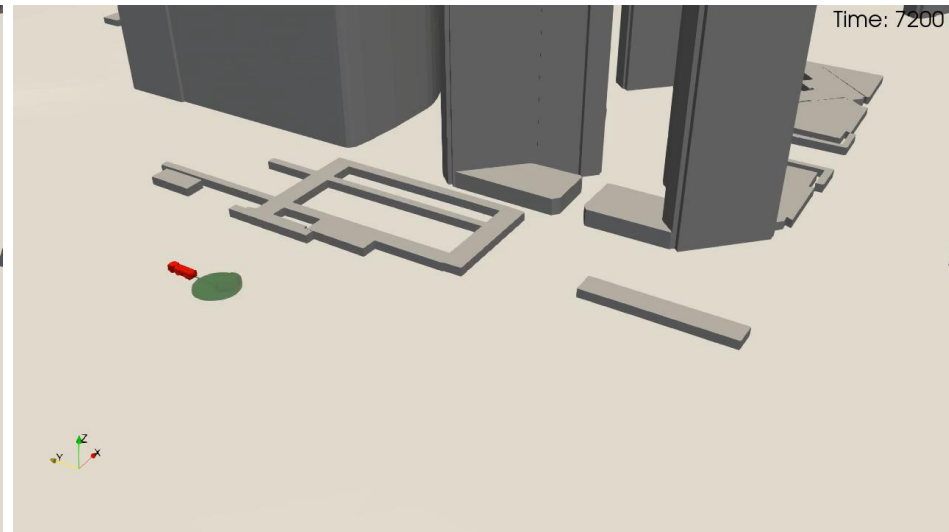
Case1



Case2



Case3



Case4

- 오픈소스 CFD 해석 라이브러리인 OpenFOAM을 활용하여, 암모니아를 실은 탱크로리의 누출에 대한 화학물질 누출 영향 범위를 예측하고자 함
- 경부선의 동탄 위치에서의 누출 사고를 가정하여 동탄 지역의 지형 및 건물 정보를 이용하여 해석 영역의 격자를 생성하였고, 동탄 지역의 평균 풍속과 풍향을 이용하여 해석 영역의 경계조건을 설정함
- 총 4가지의 누출 시나리오(기상 10분 전량 누출, 기상+액상 10분 전량 누출, 기상 누출률 고정, 기상+액상 누출률 고정)에 대한 해석 결과를 도출함
- 암모니아 농도 검출 결과는 거리에 따라 검출되는 경향이 일정하지 않고, 누출 기준 500m에서는 지형/건물 그리고 풍향의 영향으로 거리가 더 먼 지점에 비하여 낮은 농도가 검출 됨
 - 누출 기준 500m는 아파트 및 주택 단지에서 떨어진 지역이며, 누출 기준에 인접하여 누출 방향인 남쪽으로 암모니아가 퍼지는 경향때문에 낮은 농도가 검출된 것으로 판단
- 본 연구를 통해 오픈소스 소프트웨어인 OpenFOAM을 사용하여 화학물질 누출사고의 확산 범위를 해석할 수 있음을 확인하였다
 - 기존의 현상학적모델(SLAB)과 지형/건물데이터 그리고 날씨데이터를 활용을 통해 화학물질 누출사고 예측에 최적화된 모델이 개발이 용이함을 확인하였다

- 풍향
 - Dynamic Mesh 이용하여, 시간에 따른 풍향 변경 기능
- 풍속
 - 시간에 따른 풍속 변화 기능
- 지형/건물
 - 누출 지점 좌표 입력 시 자동으로 해석 영역 격자 생성하는 모듈

- OpenFOAM Guide - <https://www.openfoam.com/documentation/overview>
- FLACS Guide - <https://www.gexcon.com/support/flacs-cfd/technical-manuals/>
- 도시협곡에서의 바람유동과 오염물질 이송에 관한 수치적 연구 - 서울대학교 대학원, 건설환경공학부, 공학박사 학위 논문, Nguyen Thanh Chuyen
- 누출원 모델링에 관한 기술지침 - KOSHA GUIDE, P-92-2023, 한국산업안전보건공단
- 3-D dispersion model for simulation of accidental toxic gas releases in a metropolitan area - Journal of Loss Prevention in the Process Industries, Volume 69, March 2021, 104337, Mimi Min, Junyong Park, Chankyu Kang, Seungho Jung
a



감사합니다.