Visualisering av spredningsberegninger fra Fluent

Anders Helgeland, Geir Engdahl og Kim Kalland

Forsvarets forskningsinstitutt (FFI)

18. april 2008

FFI-rapport 2008/00695

1048

ISBN 978-82-464-1356-3

Emneord

Visualisering
Animasjon
3D data
Spredningsdata
Fluent
VoluViz

Godkjent av

Monica Endregard	Prosjektleder
Bjarne Haugstad	Forskningssjef
Jan Ivar Botnan	Avdelingssjef

Sammendrag

Dette dokumentet beskriver det arbeidet som er blitt gjort, og de stegene som må til for å visualisere tredimensjonale Fluent-data med VoluViz.

English summary

This document describes the work done and the steps necessary to visualize three-dimensional Fluent data using VoluViz.

Innhold

1 Visualisering av spredningsberegninger fra Fluent

Dagens teknologi gjør det mulig å produsere enorme mengder med data, noe som gjør at vi blir overveldet med informasjon. I mange situasjoner må forskere manuelt utforske disse datasettene for å kunne oppnå økt forståelse av et gitt fenomen.

I FFIs prosjekt 1048 “Masseødeleggelsesvåpen - trussel og beredskap” brukes programpakken Fluent for å beregne nær-feltspredning av kjemiske og radioaktive trusselstoffer. Fluent har også en rekke andre anvendelser ved FFI. Frem til nå har visualisering og analyse av tredimensjonale Fluent-data blitt utført ved bruk av standard verktøy tilgjengelig i Fluent eller ved bruk av gratis visualiseringsprogrampakker tilgjengelige på nett. Dette har vist seg å være utilstrekkelig for å kunne gjøre interaktiv analyse av store simuleringsberegninger. For å kunne oppnå dette, er det nødvendig å ta i bruk moderne visualiseringsteknikker som ikke finnes i standardpakker som brukes på FFI. Moderne visualiseringsteknikker gjør det mulig å utforske datasett på et mye større detaljnivå enn tidligere og kan sammenlignes med astronomenes bruk av teleskoper.

Problemet med bruk av avanserte visualiseringsteknikker og visualisering generelt, er at vitenskapsfolk fort kan famle i blinde ved bruk av slike verktøy uten å vite hvordan visualiseringsalgoritmen fungerer og uten å forstå vitenskapen bak visualisering [1]. Denne bakgrunnen er viktig å ha og gjør det mye enklere å nå målet med visualisering; nemlig å konvertere data til effektive og meningsfylte bilder.

Løsningen som har blitt valgt for å kunne tilfredstille det økende behovet innen visualisering og animasjon av store tredimensjonale og tidsavhengige skalarfelter, er en løsning som baserer seg på visualiseringsprogrammet VoluViz [2, 3]. Dette notatet beskriver de nødvendige stegene som er blitt gjort for å få til dette. Notatet tar spesielt for seg visualisering av innendørs og utendørs spredning av trusselstoffer [4, 5]. Å få på plass denne kapasiteten har vært grunnlaget for dette arbeidet.

2 Visualisering av Fluent-data med VoluViz

For å kunne visualisere spredningsberegninger fra Fluent ved bruk av VoluViz er følgende arbeid blitt utført:

1. **Program for Fluent-journalgenerering** - Dette programmet genererer en journalfil som forteller Fluent hva som skal skrives til fil. Filene som blir laget fungerer som inndata til programmet `fluent2hdf5`.
2. **Fluent-til-HDF5-program** - Dette programmet, `fluent2hdf5`, leser inn ustrukturerte data fra Fluent og lager et uniformt datasett i HDF5 [6] som kan leses av VoluViz.

3. **Utvidelse av VoluViz**

   (a) **Animasjonsfunksjonalitet** - VoluViz har blitt utvidet med funksjonaliteter som gjør det mulig å lage avanserte animasjoner.

   (b) **Lagring av VoluViz-scenen** - VoluViz har fått tillagt støtte for å lagre VoluViz-scenen.

   (c) **Støtte for geometri** - Støtte for visualisering av geometri hentet fra Fluent-filer er blitt lagt til VoluViz.

2.1 Program for generering av journalfiler til Fluent

Etter at Fluent-simuleringen er ferdig må all data eksporteres til ascii format. For å gjøre denne jobben mer effektiv (spesielt for tidsavhengige simuleringer) er det blitt laget et program som lager journalfiler. Disse journalfilene leses inn i Fluent og forteller Fluent hvilke filer som skal eksporteres til ascii. Filene som blir laget fungerer som inndata til et program som brukes for å konvertere Fluent-filene til det interne formatet som brukes av VoluViz (kapittel 2.2). Programmet eksekveres ved følgende kommando:

```
./create_jou_file.py pathin filetype pathout journalfile
```

, hvor

pathin	= stien til katalogen hvor Fluent-filene ligger
filetype	= etternavnet til filene som skal leses inn til Fluent
pathout	= stien til katalogen hvor Fluent skal lagre utdata
journalfile	= navnet på den journalfilen som senere skal leses av Fluent

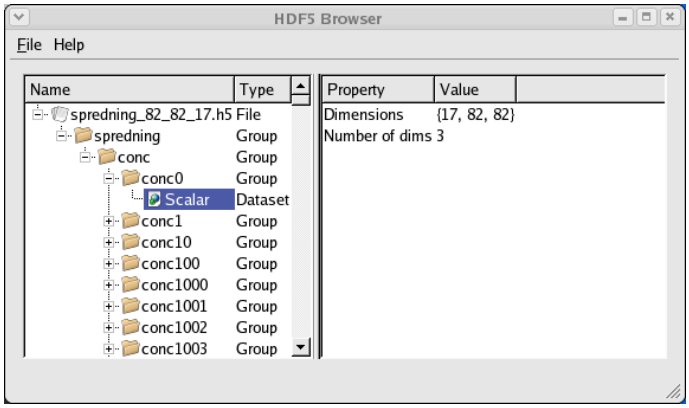
Programmet er skrevet i programmeringsspråket python og må editeres manuelt for å oppnå ønsket resultat ved ulike behov. Et eksempel på et slikt program kan finnes i appendiks A.

2.2 Fluent-til-HDF5

For å kunne bruke VoluViz, må Fluent-data først konverteres til et uniformt grid. Siden det eneste formatet som er støttet av VoluViz er HDF5, må det nye datasettet genereres i dette formatet. Programmet som er laget for å gjøre dette leser inn alle datapunktene fra Fluent-filen og genererer et nytt grid som brukeren må spesifisere som innparametere til programmet.Programmet leser først inn alle dataverdiene fra Fluent-filen og legger disse dataene i cellene til det nye griddet. Etter at dette er gjort vil celler som ikke inneholder data bli prosessert for å finne en passende dataverdi. Disse verdiene blir beregnet ut i fra de opprinnelige dataverdiene fra Fluent.

Programmet kan kjøres ved følgende kommando og parametere:

```
./fluent2hdf5 infile[basename] outfile top_group group timesteps x_dim y_dim
z_dim [min_x min_y min_z max_x max_y max_z] [samplestep]
```



Figur 2.1: HDF5 Browser.

, hvor

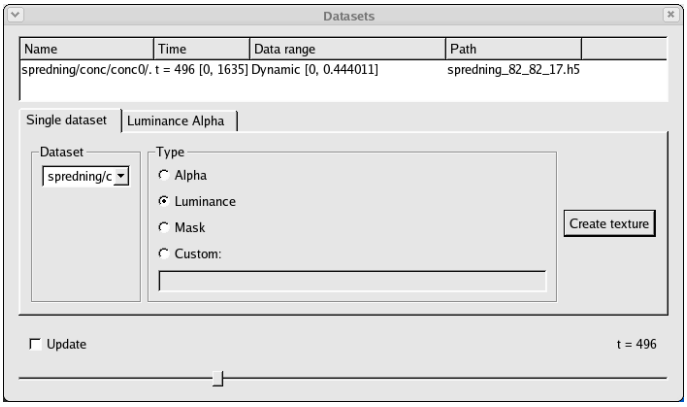
path	= stien til katalogen hvor Fluent-filene ligger
infilebasename	= basenavnet til Fluent-filene (Hvis programmet brukes til å konvertere en enkelt fil, skal det fulle navnet bli brukt)
outfile	= navnet på den nye HDF5-filen
top_group	= gruppenavnet i HDF5-filen
group	= undergruppenavnet i HDF5-filen
timesteps	= antall tidssteg i simuleringen
x_dim,y_dim,z_dim	= antall celler i x, y og z retning (i det nye griddet)
Valgfrie parametere:	
x_min,y_min,z_min	= nedre grense i fysiske koordinater
x_max,y_max,z_max	= øvre grense i fysiske koordinater (Disse kan brukes til å velge ut et mindre subset av det opprinnelige datadomenet. Hvis disse ikke er spesifisert velges det fulle domenet)
samplestep	= tidsintervallet som dataene blir "samplet" i tid (Hvis denne parameteren ikke er spesifisert blir hvert eneste tidssteg prosessert)

2.3 VoluViz

Etter at programmet beskrevet i kapittel 2.2 er kjørt, kan den nye HDF5-filen leses inn i VoluViz ved kommandoen:

```
./voluviz filename.h5
```

Etter at denne kommandoen er eksekvert har brukeren mulighet til å åpne datasett i HDF5-filen ved å klikke i 'HDF5 Browser'-dialogboksen (figur 2.1).



Figur 2.2: Datasettdialogboks.

Når brukeren har klikket på et datasett, blir datasettet lest inn internt i VoluViz. Datasettet kommer så opp i en datasettdialogboks (figur 2.2) med metainformasjon som minimum og maksimum dataverdi i tillegg til informasjon om antall tidssteg i simuleringen.

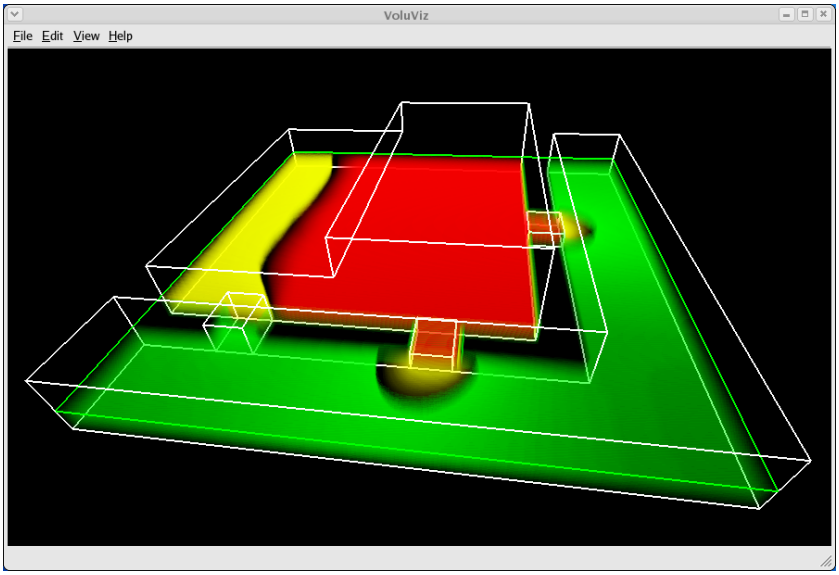
Ved å dobbelklikke på datasettet i datasettdialogboksen tegnes volumdatasettet opp i VoluViz-scenen (figur 2.3). Her kan volumet manipuleres ved bruk av tabeller for å sette farge og gjennomsiktighet samt interaktive brukerfunksjoner som rotasjon, klippeplan og valg av tidspunkt i simuleringen. Består simuleringen av flere datasett, er det også støtte for å se de samtidig.

2.3.1 Animasjon

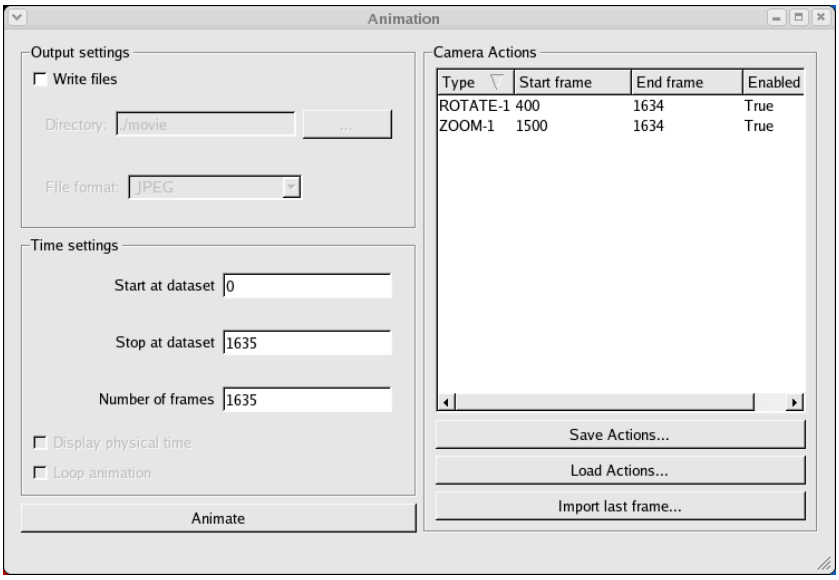
Som hjelp til å lage avanserte animasjoner er det laget en animasjonsdialogboks (figur 2.4) med mulighet til å angi start og stoppetider i tillegg til å spesifisere et sett med pregenererte kameraposisjoner. Et eksempel på bruk av ulike kameraposisjoner kan sees i figur 2.5.

2.3.2 Lagring av VoluViz-scenen

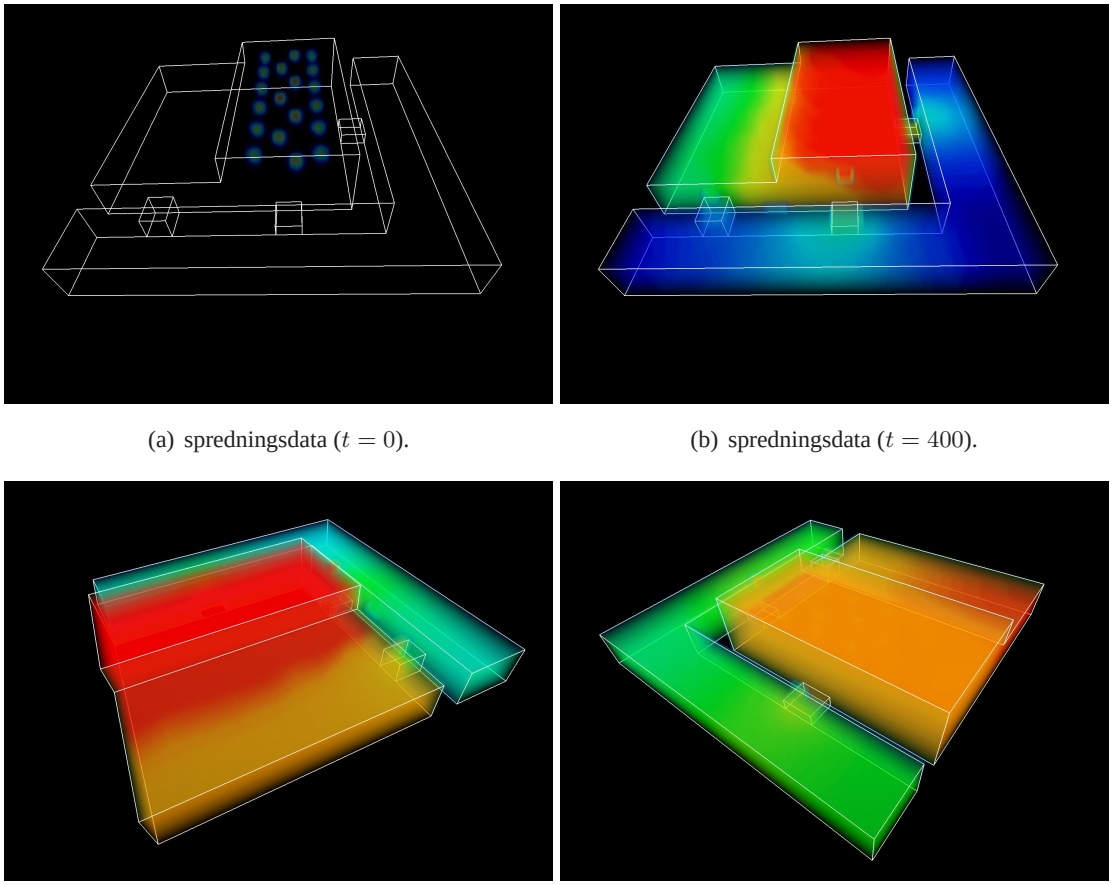
Etter at alle datasettene er lest inn, tabeller for farge og gjennomsiktighet er valgt og kameraposisjon og andre mulige parametere er blitt satt, har brukeren mulighet til å lagre hele datascenen til fil. Denne funksjonaliteten er viktig for å kunne reprodusere et arbeid. Filen som blir generert er en tekstfil som er forholdsvis enkel å lese. Dette gjør det også mulig å manipulere en VoluViz-scene kun ved å editere tekstfilen i etterkant. Et eksempel på en VoluViz-scene kan bli funnet i appendiks B.



Figur 2.3: I VoluViz-scenen kan volumdata fra simuleringsberegninger utforskes interaktivt. Denne VoluViz-scenen viser et eksempel på bruk av farge, gjennomsiktighet og klippeplan for å visualisere data fra en innendørs spredningssimulering [4].



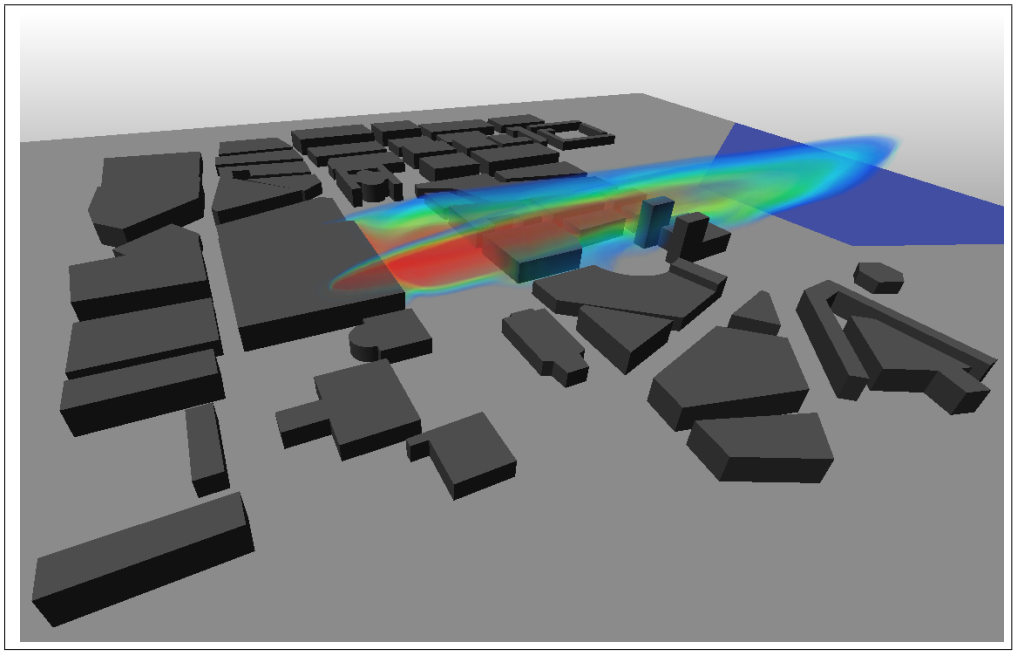
Figur 2.4: Animasjonsdialogboksen.



(a) spredningsdata ($t = 0$). (b) spredningsdata ($t = 400$).

(c) spredningsdata ($t = 800$). (d) spredingsdata ($t = 1200$).

Figur 2.5: Enkelttidspunkter hentet ut fra en animasjon av spredningsdata [4]. Frem til $t = 400$ holdes kameraposisjonen fast, mens fra $t = 400$ og ut animasjonen roterer kamera en gang rundt hele datasettet mens tiden går.



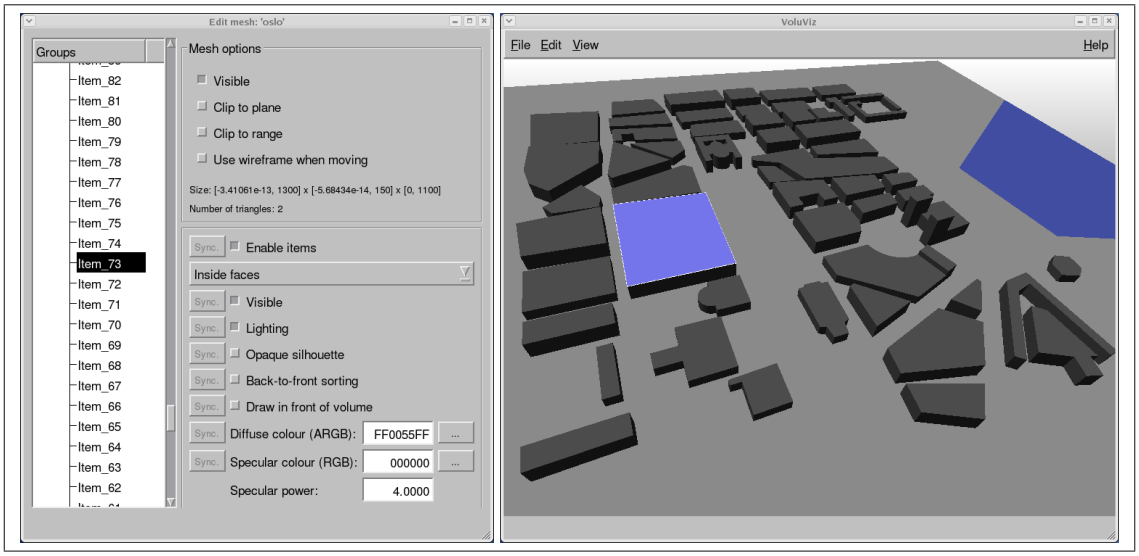
Figur 2.6: Visualisering av Fluent-geometri sammen med Fluent-data [5].

2.3.3 Geometri

For å visualisere Fluent-geometrien sammen med volumdataene (figur 2.6) er det valgt en løsning som tar utgangspunkt i (*.msh) filer som brukes av Fluent. Først må disse filene leses inn av programmet `msh2vvm`. Dette programmet leser inn mesh-filen, finner alle domenegrensene og lager en overflatebeskrivelse av Fluent-geometrien som et trekant-mesh. Programmet kjøres ved følgende kommando og opsjoner:

```
Usage: msh2vvm <input files> [options]
Options:
  -o <file>         OUTPUT file
  -p <number>       Number of simplification PASSES (default = 3)
  -a <number>       Maximum ANGLE in degrees between faces sharing normals
                    (approximate) (default = 180, 30 recommended)
  -c <number>       Maximum angle in degrees between COPLANAR faces
                    (approximate) (default = 0.25)
  -f                FLIP faces
  -n <+/-><x/y/z>   Detect surfaces with NORMAL in the given direction
  -s <number>       Minimum relative AREA a surface needs to be detected
                    (default = 0.01)
```

For å tilfredstille krav til interaktivitet, som er nødvendig for å kunne gjøre analyse av store 3D data, kan den opprinnelige Fluent-geometrien forenkles ved å kjøre programmet `msh2vvm` med passende valg av parametere. Programmet lager for eksempel forenklede versjoner av alle plane



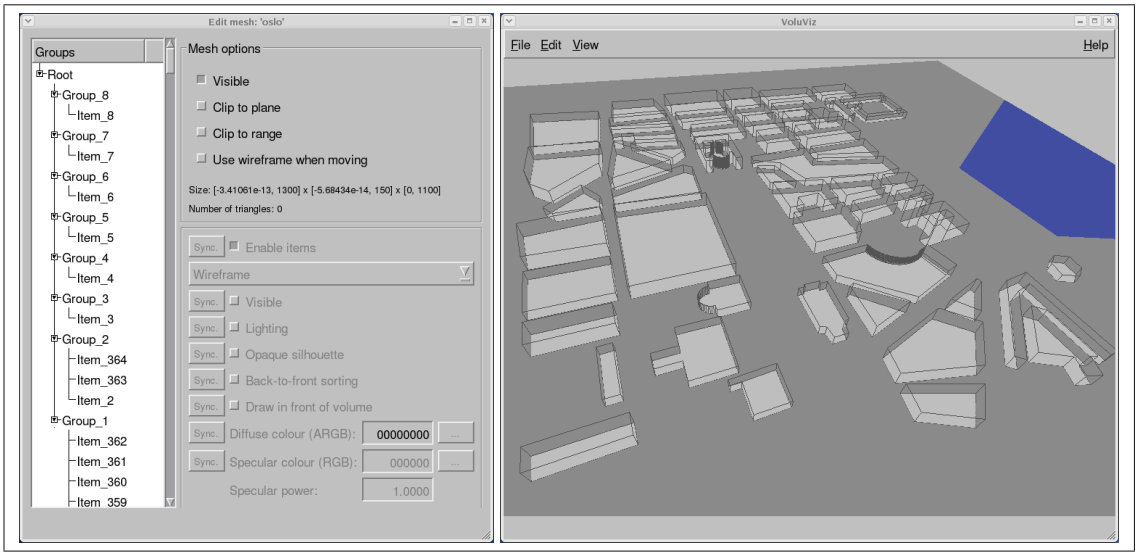
Figur 2.7: Flater kan aktiveres enten ved å velge flateelementer i mesh-editoren (venstre bilde), eller ved å klikke på de i hovedvinduet (høyre bilde). I mesh-editoren kan valg som for eksempel farge og gjennomsiktighet bli gjort for aktiverte flater (markert blått i hovedvinduet).

flater, noe som gjør at antall polygoner som trengs for å tegne opp geometrien blir kraftig redusert uten at detaljer i geometrien bli redusert.

Det nye trekant-meshet blir lagret i et internt mesh-format (*.vvm) som brukes av VoluViz. Filen kan leses inn i VoluViz ved velge **Load Mesh** i **File**-menyen. For å kunne manipulere geometrien, er det blitt laget en editor hvor brukeren har mulighet til å klikke på flater for å velge farge og gjennomsiktighet (figur 2.7). Det er også lagt til støtte for å vise hele eller deler av geometrien i type “wireframe”-representasjon (figur 2.8).

3 Oppsumering

Ved dette arbeidet har en rekke verktøy blitt laget som gjør det mulig å visualisere Fluent-data. Disse verktøyene kan brukes både til å gjøre analyse av simuleringsdata samt å lage avanserte animasjoner som kan brukes i presentasjonssammenhenger. Arbeid som kan gjøres i fremtiden er å lage støtte for andre størrelser enn skalarer, og basere innlesingen av Fluent-filene på et annet format enn ascii. Disse filene blir fort veldig store og er ineffektive å jobbe med.



Figur 2.8: Visualisering av Fluent-geometri som "wireframe".

Referanser

[1] C. Johnson, "Top Scientific Visualization Research Problems," *IEEE Comput. Graph. Appl.*, vol. 24, no. 4, pp. 13–17, 2004.

[2] T. Gaarder and A. Helgeland, "VoluViz 1.0 Report," FFI (Norwegian Defence Research Establishment), Tech. Rep. FFI/RAPPORT-2002/03449, Sept. 2002.

[3] G. Engdahl and A. Helgeland, "VoluViz 1.3 Report," FFI (Norwegian Defence Research Establishment), Tech. Rep. FFI/RAPPORT-2005/02506, 2005.

[4] M. Endregard, B. A. P. Reif, and T. Vik, "Consequence assessment of the indoor dispersion of sarin," FFI (Norwegian Defence Research Establishment), Tech. Rep. FFI/RAPPORT-2002/XXX, 2008, in press.

[5] M. Lindén and E. Wingsten, "Cfd modelling of urban dispersion of aerosols," Master's thesis, Dept. Applied Mechanics, Div. Fluid Dynamics, Chalmers University of Technology, Gothenburg, Sweden, 2007, ISSN 1652-8557.

[6] "HDF5," 2007. [Online]. Available: http://hdf.ncsa.uiuc.edu/HDF5/

A Program for Fluent-journalgenerering

```
#!/usr/bin/env python2
#####!/usr/freeware/bin/python

import sys
import os
import string
import re

if len(sys.argv) != 5:
    print "./app.py pathin filetype pathout journalfile"
    sys.exit()

pathin = sys.argv[1]
ftype = sys.argv[2]
pathout = sys.argv[3]
fileout = sys.argv[4]

ftype = "." + ftype

flist = []
for file in os.listdir(pathin):
    if (file.endswith(ftype)):
        flist.append(file)

pat = re.compile('([^0-9]*)([0-9]*)')

nlist = []
s = ""
for f in flist:
    ps = pat.search(f)
    if ps:
        s, num = ps.group(1, 2)
        if num == "":
            print 'File %s: Not according to supported format' % f
        else:
            nlist.append(string.atoi(num))
    else:
        print 'no match'

nlist.sort()

ftypeout = ".ascii"


# Generate input fluent file
file = open(fileout, 'w')


# iterate files
i = 0;
for n in nlist:
    # insert beginning zeros (ex: 20 -> 0020)
    if (n < 10):
```

```
        filein = s + '000' + str(n) + ftype
    elif (n < 100):
        filein = s + '00' + str(n) + ftype
    elif (n < 1000):
        filein = s + '0' + str(n) + ftype
    else:
        filein = s + str(n) + ftype

    fileout = s + str(i) + ftypeout
    i = i + 1
    if os.path.isfile(pathin + "/" + filein):

        ##############################################
        ## Kommandoene under forteller Fluent hvilke
        ## filer som skal prosesseres og hva som skal
        ## gjøres med de.
        ## Denne delen av koden kan editeres etter
        ## behov
        ##############################################
        file.write('/file\n')
        file.write('read-data\n')
        file.write('"' + pathin + "/" + filein + '"' + '\n')
        file.write('export\n')
        file.write('ascii\n')
        file.write('"' + pathout + "/" + fileout + '"' + '\n')
        file.write('()\n')
        file.write('no\n')
        file.write('no\n')
        file.write('uds-0-scalar\n')
        file.write('q\n')
        file.write('no\n')
        file.write('q\n')
        file.write('q\n')
        file.write('q\n')
        file.write('\n')
        ##############################################

    else:
        "no such file"

file.close
```

B VoluViz-scenefil

```
<vvDatasetManager>
<vvDataset datasetName="Scalar" fileName="path/filename1.h5" groupName="groupname1_492/."
key="groupname0/.">
<vvDatasetRange max="1.5" min="0">
</vvDataset>
```

```
<vvDataset datasetName="Scalar" fileName="path/filename2.h5" groupName="groupname2_1635/."
key="groupname2_0/.">
<vvDatasetRange max="0.23" min="0">
</vvDataset>
</vvDatasetManager>
<vvVisObjectManager>
<vvTMVisObject TMVisObjectType="LUMINANCE" lumName="dosage0/." shaderType="L">
<vvColorTable colormaptype="RGBA" colorspace="HSVA" comptype="FLOAT" magic="33637"
numindices="256" version="2">
<vvColorTableChannel channel="2" interpolation="linear" numpoints="7">
<vvColorTablePoint index="0" value="0.331865">
<vvColorTablePoint index="16" value="0.339119">
<vvColorTablePoint index="91" value="0.334091">
<vvColorTablePoint index="92" value="0.165026">
<vvColorTablePoint index="211" value="0.175">
<vvColorTablePoint index="212" value="0">
<vvColorTablePoint index="255" value="0">
</vvColorTableChannel>
<vvColorTableChannel channel="3" interpolation="linear" numpoints="2">
<vvColorTablePoint index="0" value="1">
<vvColorTablePoint index="255" value="1">
</vvColorTableChannel>
<vvColorTableChannel channel="4" interpolation="linear" numpoints="13">
<vvColorTablePoint index="0" value="0">
<vvColorTablePoint index="15" value="0">
<vvColorTablePoint index="16" value="0">
<vvColorTablePoint index="19" value="1">
<vvColorTablePoint index="90" value="1">
<vvColorTablePoint index="91" value="0">
<vvColorTablePoint index="92" value="0">
<vvColorTablePoint index="95" value="1">
<vvColorTablePoint index="211" value="1">
<vvColorTablePoint index="212" value="0">
<vvColorTablePoint index="213" value="0">
<vvColorTablePoint index="216" value="1">
<vvColorTablePoint index="255" value="1">
</vvColorTableChannel>
<vvColorTableChannel channel="5" interpolation="linear" numpoints="9">
<vvColorTablePoint index="0" value="0">
<vvColorTablePoint index="15" value="0">
<vvColorTablePoint index="86" value="0.111364">
<vvColorTablePoint index="87" value="0">
<vvColorTablePoint index="91" value="0">
<vvColorTablePoint index="205" value="0.127273">
<vvColorTablePoint index="208" value="0">
<vvColorTablePoint index="212" value="0">
<vvColorTablePoint index="255" value="0.0853468">
</vvColorTableChannel>
</vvColorTable>
</vvTMVisObject>
<vvTMVisObject TMVisObjectType="LUMINANCE" lumName="varemessa/C/C0/." shaderType="L">
<vvColorTable colormaptype="RGBA" colorspace="HSVA" comptype="FLOAT" magic="33637"
numindices="256" version="2">
<vvColorTableChannel channel="2" interpolation="linear" numpoints="8">
<vvColorTablePoint index="0" value="0.715909">
<vvColorTablePoint index="28" value="0.493668">
<vvColorTablePoint index="62" value="0.408079">
<vvColorTablePoint index="105" value="0.169651">
```

```
<vvColorTablePoint index="150" value="0.126856">
<vvColorTablePoint index="192" value="0.0962882">
<vvColorTablePoint index="206" value="0">
<vvColorTablePoint index="255" value="0">
</vvColorTableChannel>
<vvColorTableChannel channel="3" interpolation="linear" numpoints="2">
<vvColorTablePoint index="0" value="1">
<vvColorTablePoint index="255" value="1">
</vvColorTableChannel>\rule{\textwidth}{1pt}

<vvColorTableChannel channel="4" interpolation="linear" numpoints="3">
<vvColorTablePoint index="0" value="1">
<vvColorTablePoint index="202" value="1">
<vvColorTablePoint index="255" value="1">
</vvColorTableChannel>
<vvColorTableChannel channel="5" interpolation="linear" numpoints="3">
<vvColorTablePoint index="0" value="0">
<vvColorTablePoint index="9" value="0.0255294">
<vvColorTablePoint index="255" value="0.0255294">
</vvColorTableChannel>
</vvColorTable>
</vvTMVisObject>
</vvVisObjectManager>
<vvViewer aspect="1.13214" axis="0" bbox="0" clip="0" dolly="1.92933" fov="60" reduce="1">
<vvBgColor blue="0" corner="0" green="0" red="0">
<vvBgColor blue="0" corner="1" green="0" red="0">
<vvBgColor blue="0" corner="2" green="0" red="0">
<vvBgColor blue="0" corner="3" green="0" red="0">
<vvROI xMax="81" xMin="0" yMax="81" yMin="0" zMax="5" zMin="0">
<vvVirtualSphereController role="eye" rot1="-0.676862" rot2="0.336229" rot3="-0.271584"
rot4="-0.59586"
xTranslation="0" yTranslation="0" zTranslation="0">
<vvVirtualSphereController role="clip plane" rot1="2.83268e-05" rot2="0" rot3="0" rot4="1"
xTranslation="-0.0841219" yTranslation="0.304762" zTranslation="0">
<vvManipulator value="MAN_SHAPE">
<vvSampleRate role="standard" x="5" y="5" z="5">
<vvSampleRate role="reduced" x="0.5" y="0.5" z="0.5">
<vvVolumeScale x="1" y="1" z="1">
<vvVolumeTranslation x="-0.0333333" y="0.606667" z="0">
<vvClipPlaneTranslation x="0.000710987" y="0" z="-5.05905">
</vvViewer>
<vvAnimation numFrames="1635" timeBegin="0" timeEnd="1635">
<vvRotation angleBegin="0" angleEnd="360" enabled="true" frameBegin="400" frameEnd="1634"
xAxisBegin="0" xAxisEnd="0" yAxisBegin="0" yAxisEnd="0" zAxisBegin="1" zAxisEnd="1">
</\rule{\textwidth}{1pt}vvAnimation>
<vvMainWindow height="888" width="951" x="0" y="26">
```