Bavarian Graduate School of Computational Engineering

Technische Universität München

Installation Guide

CADTOPCAD-a CAD to Optimized Topology to CAD Software Tool

Authors: Saumitra Joshi,
Juan Carlos Medina,
Friedrich Menhorn,
Severin Reiz,
Benjamin Rüth,
Erik Wannerberg,
Anna Yurova

Advisors: Arash Bakhtiari (TUM),
Dirk Hartmann (Siemens AG),
Utz Wever (Siemens AG)

Preface

The Bavarian Graduate School of Computational Engineering (BGCE) honours project at the Computational Science and Engineering (CSE) Institute of Technische Universität München (TUM) is a 10-month project where students conduct research on cutting-edge topics in the field of Computational Engineering, in cooperation with a partner in industry or academia. The 2015–16 project is titled *CAD-Integrated Topology Optimization* and is initiated and supervised in a cooperation between TUM and Siemens AG in Munich.

Todo list

Contents

1 Introduction

In this document we provide a complete guide on how to install and use CADTOPCAD tool on Linux.

2 ToPy

In our tool we use ToPy (https://github.com/williamhunter/topy) for topology optimization.

2.1 Prerequisites

In order to install ToPy, make sure that the following software is installed on your computer:

- Python (version > 2.7)
- NumPy (Usually provided by Python distribution)
- PyVTK tool (https://pypi.python.org/pypi/PyVTK)
- Pysparse library(http://pysparse.sourceforge.net/)

Here are some recommendation for the installation of the above mentioned tools/libraries.
To install PyVTK tool, please run the following commands in your terminal:

```
sudo apt-get install python-pip
sudo pip install pyvtk
```

The installation of the Pysparse library is a little bit more cumbersome, since the pip-installation (like in the previous case) fails most of the times. So, here we provide an alternative way of installing Pysparse from the *.git* repository.
To install Pysparse, make sure that *git* (url) is installed on your computer and then run the following commands in your terminal:

```
git clone git://pysparse.git.sourceforge.net/gitroot/pysparse/
   pysparse/
\item cd pysparse
\item sudo python setup.py install
```

Furthermore, for CADTOPCAD it is necessary to have an output in the *ascii* format. By default the output *.vtk* files from ToPy are binary, so we need to change them to *ascii*. In order to do that, please perform the following actions:

- Open the ToPy source file `core/visualization.py`
- Go to the method `_write_legace_vtu(x, fname)` (line 160)
- Change in line 194 `binary` to `ascii` (see pic. 2.1.1)

```python

def _write_legacy_vtu(x, fname):
    """
    Write a legacy VTK unstructured grid file.

    """
    # Voxel local points relative to its centre of geometry:
    voxel_local_points = asarray([[-1,-1,-1],[ 1,-1,-1],[-1, 1,-1],[ 1, 1,-1],
                                  [-1,-1, 1],[ 1,-1, 1],[-1, 1, 1],[ 1, 1, 1]])\
                                    * 0.5 # scaling
    # Voxel world points:
    points = []
    # Culled input array -- as list:
    xculled = []

    try:
        depth, rows, columns = x.shape
    except ValueError:
        sys.exit('Array dimensions not equal to 3, possibly 2-dimensional.\n')

    for i in xrange(depth):
        for j in xrange(rows):
            for k in xrange(columns):
                if x[i,j,k] > THRESHOLD:
                    xculled.append(x[i,j,k])
                    points += (voxel_local_points + [i,j,k]).tolist()

    voxels = arange(len(points)).reshape(len(xculled), 8).tolist()
    topology = UnstructuredGrid(points, voxel = voxels)
    file_header = \
    'ToPy data, created '\
    + str(datetime.now()).rsplit('.')[0]
    scalars = CellData(Scalars(xculled, name='Densities', lookup_table =\
    'default'))
    vtk = VtkData(topology, file_header, scalars)
    vtk.tofile(fname, 'ascii')

def  timestamp():
```

Figure 2.1.1: Changing of the output type of ToPy to ascii

2.2 Install ToPy

If all the tools specified in the section 2.1 are installed, we can now proceed to the installation of ToPy itself. For that download ToPy from https://github.com/williamhunter/topy and run the following command from the root directory of ToPy:

Did we clone it from git or downloaded it?

```
sudo python setup.py install
```

```
                    python optimise.py cantilvr_3d_etaopt_gsf.tpd

==========================================================================
ToPy problem definition (TPD) file successfully parsed.
TPD file name: cantilvr_3d_etaopt_gsf.tpd (v2007)
--------------------------------------------------------------------------
Domain discretisation (NUM_ELEM_X x NUM_ELEM_Y x NUM_ELEM_Z) = 28 x 37 x 111
Element type (ELEM_K) = H8
Filter radius (FILT_RAD) = 1.5
Number of iterations (NUM_ITER) = 50
Problem type (PROB_TYPE) = comp
Problem name (PROB_NAME) = cantilvr_3d_etaopt_gsf
GSF active
Damping factor (ETA) = 0.40
No passive elements (PASV_ELEM) specified
Active elements (ACTV_ELEM) specified
==========================================================================
 Iter | Obj. func.   | Vol.  | Change     | P_FAC | Q_FAC | Ave ETA | S-V fra
--------------------------------------------------------------------------
ToPy: Solution for FEA converged after 451 iterations.
  1  | 5.231335e+01 | 0.150 | 8.5000e-01 | 1.000 | 1.000 |  0.400  |  0.010
ToPy: Solution for FEA converged after 452 iterations.
  2  | 3.023124e+01 | 0.150 | 2.0000e-01 | 1.000 | 1.000 |  0.400  |  0.010
ToPy: Solution for FEA converged after 499 iterations.
  3  | 2.455771e+01 | 0.150 | 2.0000e-01 | 1.000 | 1.000 |  0.400  |  0.012
ToPy: Solution for FEA converged after 540 iterations.
  4  | 2.190561e+01 | 0.150 | 2.0000e-01 | 1.000 | 1.000 |  0.400  |  0.014
ToPy: Solution for FEA converged after 562 iterations.
  5  | 2.038384e+01 | 0.150 | 1.5306e-01 | 1.000 | 1.000 |  0.400  |  0.019
ToPy: Solution for FEA converged after 569 iterations.
  6  | 1.987525e+01 | 0.150 | 1.2965e-01 | 1.000 | 1.000 |  0.400  |  0.023
ToPy: Solution for FEA converged after 570 iterations.
  7  | 1.968465e+01 | 0.150 | 7.9084e-02 | 1.000 | 1.000 |  0.400  |  0.027
ToPy: Solution for FEA converged after 564 iterations.
  8  | 1.956767e+01 | 0.150 | 5.6287e-02 | 1.000 | 1.000 |  0.400  |  0.030
ToPy: Solution for FEA converged after 571 iterations.
```

Figure 2.2.1: ToPy test

2.3 Test ToPy

In order to test whether the installation of ToPy was completed successfully it is possible to run some test cases provided in *examples* folder. For that, do the following:

- Enter one of the folders in examples (e.g. examples/cantilever)
- Execute a ToPy test run by running the following command in your terminal:

  ```
  python optimize.py <example.tpd-file>
  ```

The output should look as showed on a picture 2.2.1.

3 OpenCascade

OpenCascade (http://www.opencascade.com/) is a...

maybe add a short description

3.1 Install OpenCascade

For technical reasons, we do not use OpenCascade from the official webpage, but from the *.git* repository. To install OpenCascade this way, make sure that textitgit (url) is installed on your computer and then run the following commands in your terminal:

- ```
  git clone git://github.com/tpaviot/oce.git
  cd oce
  mkdir build
  cd build
  ```

- `cmake ..`

  Sample output: see Pic. 3.1.1

- `make ..`

  Sample output: see Pic. 3.1.2

- `sudo make install ..`

  Sample output: see Pic. 3.1.3

Figure 3.1.1: Building OpenCascade
  ```

```
File  Edit  View  Search  Terminal  Help
-- Processing ToolKit: TKStdLSchema (StdLSchema;StdLDrivers)
-- Processing ToolKit: TKCAF (TDataXtd;TNaming;TPrsStd;AppStd)
-- Processing ToolKit: TKBin (BinDrivers;BinMDataXtd;BinMPrsStd;BinMNaming;BinTools)
-- Processing ToolKit: TKXml (XmlDrivers;XmlMDataXtd;XmlMNaming;XmlMPrsStd)
-- Processing ToolKit: TKPCAF (PDataXtd;PNaming;PPrsStd;MDataXtd;MPrsStd;MNaming)
-- Processing ToolKit: TKBinTObj (BinTObjDrivers)
-- Processing ToolKit: TKXmlTObj (XmlTObjDrivers)
-- Processing ToolKit: TKStdSchema (StdSchema;StdDrivers)
-- Processing ToolKit: TKSTL (StlMesh;StlAPI;StlTransfer;RWStl)
-- Processing ToolKit: TKXSBase (Interface;Transfer;IFGraph;IFSelect;TransferBRep;XSControl;StepData;StepFile;HeaderSection;RWHeaderSectio
n;APIHeaderSection;StepSelect;UnitsMethods;XSAlgo;LibCtl;MoniTool)
-- Processing ToolKit: TKSTEPBase (StepBasic;RWStepBasic;StepRepr;RWStepRepr;StepGeom;RWStepGeom;StepShape;RWStepShape)
-- Processing ToolKit: TKIGES (IGESData;IGESFile;IGESBasic;IGESGraph;IGESGeom;IGESDimen;IGESDraw;IGESSolid;IGESDefs;IGESAppli;IGESConvGeom
;IGESSelect;IGESToBRep;GeomToIGES;Geom2dToIGES;BRepToIGES;BRepToIGESBRep;IGESControl)
-- Processing ToolKit: TKSTEPAttr (StepVisual;RWStepVisual;StepDimTol;RWStepDimTol)
-- Processing ToolKit: TKSTEP209 (StepElement;StepFEA;RWStepElement;RWStepFEA)
-- Processing ToolKit: TKSTEP (StepAP214;RWStepAP214;StepAP203;RWStepAP203;STEPConstruct;STEPEdit;GeomToStep;StepToGeom;StepToTopoDS;TopoD
SToStep;STEPControl;STEPSelections;StepAP209)
-- Processing ToolKit: TKVRML (VrmlConverter;VrmlAPI;Vrml;VrmlData)
-- Processing ToolKit: TKXCAF (XCAFApp;XCAFDoc;XCAFPrs)
-- Processing ToolKit: TKXCAFSchema (MXCAFDoc;PXCAFDoc;XCAFDrivers;XCAFSchema)
-- Processing ToolKit: TKXmlXCAF (XmlXCAFDrivers;XmlMXCAFDoc)
-- Processing ToolKit: TKBinXCAF (BinXCAFDrivers;BinMXCAFDoc)
-- Processing ToolKit: TKXDEIGES (IGESCAFControl)
-- Processing ToolKit: TKXDESTEP (STEPCAFControl)
-- Processing ToolKit: TKDraw (Draw;DBRep;DrawTrSurf)
-- Processing ToolKit: TKTopTest (TestTopOpeDraw;TestTopOpeTools;TestTopOpe;BRepTest;GeometryTest;HLRTest;MeshTest;GeomliteTest;DrawFairCu
rve;BOPTest;SWDRAW)
-- Processing ToolKit: TKViewerTest (ViewerTest)
-- Processing ToolKit: TKXSDRAW (SWDRAW;XSDRAW;XSDRAWIGES;XSDRAWSTEP;XSDRAWSTLVRML)
-- Processing ToolKit: TKDCAF (DDF;DDocStd;DNaming;DDataStd;DPrsStd;DrawDim)
-- Processing ToolKit: TKXDEDRAW (XDEDRAW)
-- Processing ToolKit: TKTObjDRAW (TObjDRAW)
-- Processing application: DRAWEXE (DRAWEXE)
-- Configuring done
-- Generating done
-- Build files have been written to:                                git/oce-master/build
```

Figure 3.1.2: OpenCascade installation: cmake

```
File  Edit  View  Search  Terminal  Help
-- Installing: /usr/local/include/oce/TestTopOpeDraw_TTOT.hxx
-- Installing: /usr/local/include/oce/TestTopOpeTools_Trace.hxx
-- Installing: /usr/local/include/oce/TestTopOpe_BOOP.hxx
-- Installing: /usr/local/include/oce/TestTopOpe_VarsTopo.hxx
-- Installing: /usr/local/include/oce/TestTopOpe_HDSDisplayer.hxx
-- Installing: /usr/local/include/oce/HLRTest_ShapeData.lxx
-- Installing: /usr/local/include/oce/HLRTest_OutLiner.lxx
-- Installing: /usr/local/include/oce/HLRTest_DrawablePolyEdgeTool.lxx
-- Installing: /usr/local/include/oce/HLRTest_Projector.lxx
-- Installing: /usr/local/include/oce/MeshTest_CheckTopology.hxx
-- Installing: /usr/local/include/oce/MeshTest_DrawableMesh.hxx
-- Installing: /usr/local/include/oce/BOPTest_Chronometer.hxx
-- Installing: /usr/local/lib/oce-0.17-dev/libTKTopTest.so.10.0.0
-- Installing: /usr/local/lib/oce-0.17-dev/libTKTopTest.so.10
-- Installing: /usr/local/lib/oce-0.17-dev/libTKTopTest.so
-- Set runtime path of "/usr/local/lib/oce-0.17-dev/libTKTopTest.so.10.0.0" to "/usr/local/lib/oce-0.17-dev:/usr/local/lib"
-- Installing: /usr/local/include/oce/ViewerTest_AutoUpdater.hxx
-- Installing: /usr/local/include/oce/ViewerTest_EventManager.lxx
-- Installing: /usr/local/lib/oce-0.17-dev/libTKViewerTest.so.10.0.0
-- Installing: /usr/local/lib/oce-0.17-dev/libTKViewerTest.so.10
-- Installing: /usr/local/lib/oce-0.17-dev/libTKViewerTest.so
-- Set runtime path of "/usr/local/lib/oce-0.17-dev/libTKViewerTest.so.10.0.0" to "/usr/local/lib/oce-0.17-dev:/usr/local/lib"
-- Installing: /usr/local/include/oce/XSDRAW_Commands.hxx
-- Installing: /usr/local/lib/oce-0.17-dev/libTKXSDRAW.so.10.0.0
-- Installing: /usr/local/lib/oce-0.17-dev/libTKXSDRAW.so.10
-- Installing: /usr/local/lib/oce-0.17-dev/libTKXSDRAW.so
-- Set runtime path of "/usr/local/lib/oce-0.17-dev/libTKXSDRAW.so.10.0.0" to "/usr/local/lib/oce-0.17-dev:/usr/local/lib"
-- Installing: /usr/local/include/oce/DDF_IOStream.hxx
-- Installing: /usr/local/include/oce/DDF_AttributeBrowser.hxx
-- Installing: /usr/local/include/oce/ModelDefinitions.hxx
-- Installing: /usr/local/lib/oce-0.17-dev/libTKDCAF.so.10.0.0
-- Installing: /usr/local/lib/oce-0.17-dev/libTKDCAF.so.10
-- Installing: /usr/local/lib/oce-0.17-dev/libTKDCAF.so
-- Set runtime path of "/usr/local/lib/oce-0.17-dev/libTKDCAF.so.10.0.0" to "/usr/local/lib/oce-0.17-dev:/usr/local/lib"
-- Installing: /usr/local/lib/oce-0.17-dev/libTKXDEDRAW.so.10.0.0
-- Installing: /usr/local/lib/oce-0.17-dev/libTKXDEDRAW.so.10
-- Installing: /usr/local/lib/oce-0.17-dev/libTKXDEDRAW.so
```

Figure 3.1.3: OpenCascade installation

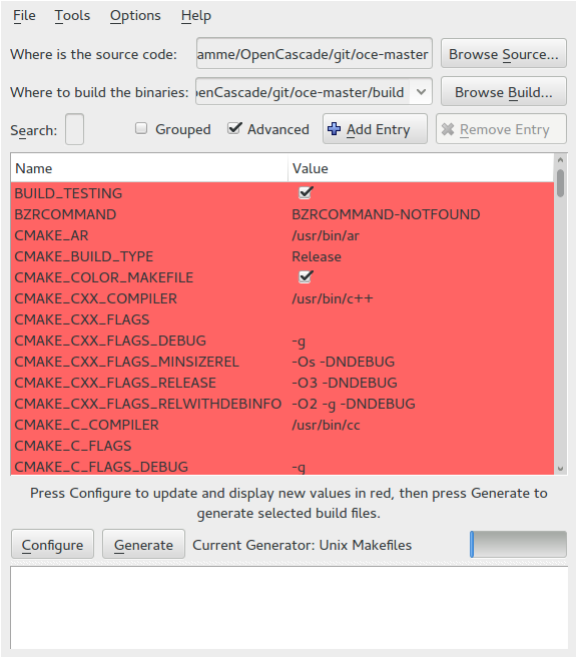
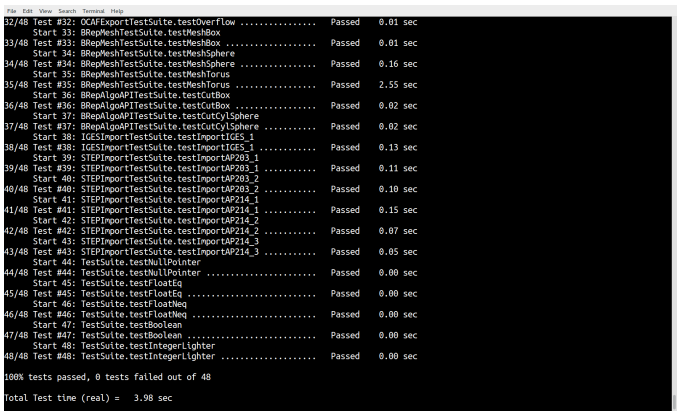


Figure 3.1.4: CMake graphical interface

In the make step, one can use the `-jx` parameter, where x is the number of processors, to build in parallel. That allows to speed up the installation process These steps are in accord with the installation guide on the git page itself. One can also use the CMake-GUI (see Pic. 3.1.4) to change some of the build configuration if need be (e.g. include OpenMP support).

3.2 Test OpenCascade

In order to test whether the installation of OpenCascade was completed successfully it is possible to run a test provided by OpenCascade.



```
File  Edit  View  Search  Terminal  Help
32/48 Test #32: OCAFExportTestSuite.testOverflow ...............   Passed    0.01 sec
      Start 33: BRepMeshTestSuite.testMeshBox
33/48 Test #33: BRepMeshTestSuite.testMeshBox ..................   Passed    0.01 sec
      Start 34: BRepMeshTestSuite.testMeshSphere
34/48 Test #34: BRepMeshTestSuite.testMeshSphere ...............   Passed    0.16 sec
      Start 35: BRepMeshTestSuite.testMeshTorus
35/48 Test #35: BRepMeshTestSuite.testMeshTorus ................   Passed    2.55 sec
      Start 36: BRepAlgoAPITestSuite.testCutBox
36/48 Test #36: BRepAlgoAPITestSuite.testCutBox ................   Passed    0.02 sec
      Start 37: BRepAlgoAPITestSuite.testCutCylSphere
37/48 Test #37: BRepAlgoAPITestSuite.testCutCylSphere ..........   Passed    0.02 sec
      Start 38: IGESImportTestSuite.testImportIGES_1
38/48 Test #38: IGESImportTestSuite.testImportIGES_1 ...........   Passed    0.13 sec
      Start 39: STEPImportTestSuite.testImportAP203_1
39/48 Test #39: STEPImportTestSuite.testImportAP203_1 ..........   Passed    0.11 sec
      Start 40: STEPImportTestSuite.testImportAP203_2
40/48 Test #40: STEPImportTestSuite.testImportAP203_2 ..........   Passed    0.10 sec
      Start 41: STEPImportTestSuite.testImportAP214_1
41/48 Test #41: STEPImportTestSuite.testImportAP214_1 ..........   Passed    0.15 sec
      Start 42: STEPImportTestSuite.testImportAP214_2
42/48 Test #42: STEPImportTestSuite.testImportAP214_2 ..........   Passed    0.07 sec
      Start 43: STEPImportTestSuite.testImportAP214_3
43/48 Test #43: STEPImportTestSuite.testImportAP214_3 ..........   Passed    0.05 sec
      Start 44: TestSuite.testNullPointer
44/48 Test #44: TestSuite.testNullPointer ......................   Passed    0.00 sec
      Start 45: TestSuite.testFloatEq
45/48 Test #45: TestSuite.testFloatEq ..........................   Passed    0.00 sec
      Start 46: TestSuite.testFloatNeq
46/48 Test #46: TestSuite.testFloatNeq .........................   Passed    0.00 sec
      Start 47: TestSuite.testBoolean
47/48 Test #47: TestSuite.testBoolean ..........................   Passed    0.00 sec
      Start 48: TestSuite.testIntegerLighter
48/48 Test #48: TestSuite.testIntegerLighter ...................   Passed    0.00 sec

100% tests passed, 0 tests failed out of 48

Total Test time (real) =   3.98 sec
```

Figure 3.2.1: OpenCascade test

For that, run the following command from your terminal:

```
make test
```

All performed tests should be successful (See Pic. 3.2.1)

4 CADTOPCAD

4.1 Prerequisites

In order to install CADTOPCAD the following tools should be installed on your computer:

- Topy (see Sec. 2)
- OpenCascade (see Sec. 3)
- (CPPUnit)

In order to install CPPUnit run the following command from you terminal:

```
sudo apt-get install lib-cppunitdev
```