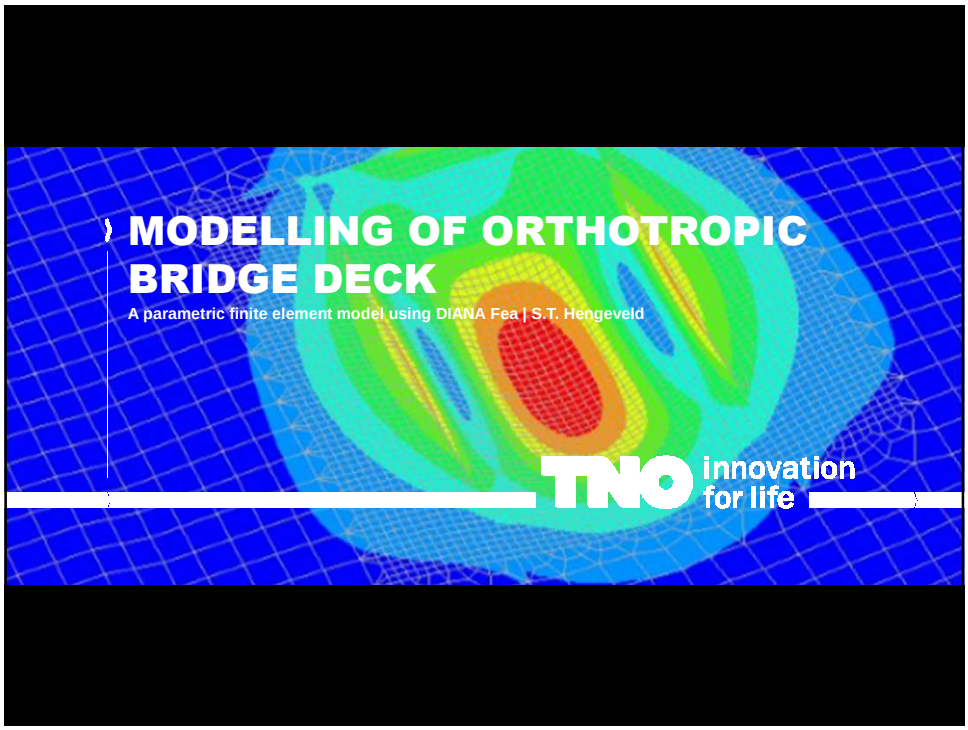
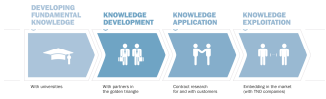


INTRODUCTION TO TNO

- 45 Departments
- 9 Units
- Buildings - Infrastructures and Maritime
  - Structural reliability
  - Structural dynamics
  - Innovation centre for building
- Products and services
  - Modelling and simulation studies
  - On-site and offshore measurements
  - Laboratory experiments: fatigue, fracture, shock

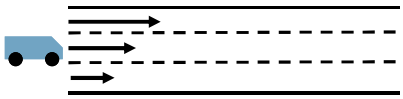


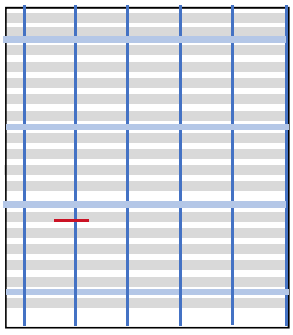
INTRODUCTION

- The Netherlands has a lot of steel bridges
- Changing traffic loads
- Fatigue life time is important
- Orthotropic bridge
  - Transversal and longitudinal girders
  - Troughs
  - Deckplate
- Focus of this presentation is the interface between python and Diana

Hartelbrug, Royal HaskoningDHV

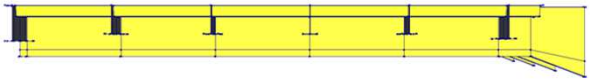
SHOWCASE





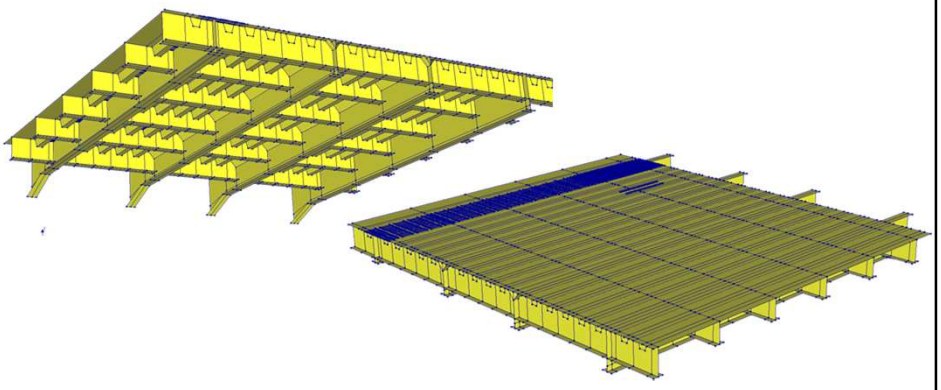
- A single bridge owned by the community
- A two lane bridge + Entrance lane
- Four longitudinal girders
- Six transversal girders
- Seventeen troughs
- Goal: Determine maintenance and inspections intervals
  - Critical location Trough-Deck connection
  - Strain range → Stress range → Fatigue life time → Maintenance interval

FEA MODEL

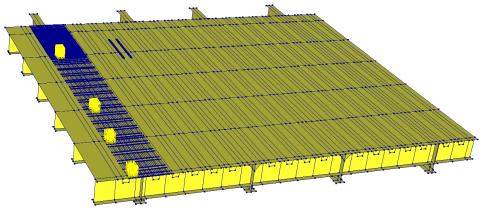


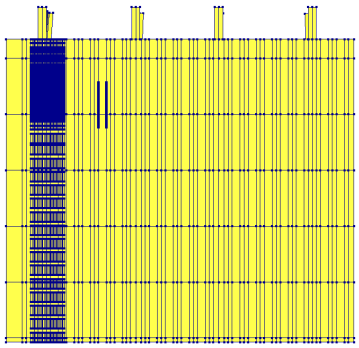
- Build in DIANA 10.1
- Shell elements
  - Quadratic
  - 200 000 ~ 300 000 Elements
- Surface loads to represent wheelprints
  - Over 3000 different locations calculated
  - Mesh refined below wheelprints and near POI's
- Linear elastic analyse
  - Super position of results to find axle loads
- Validated model with measurements
  - Strain gauges at bottom side of trought

OVERVIEW OF MODEL



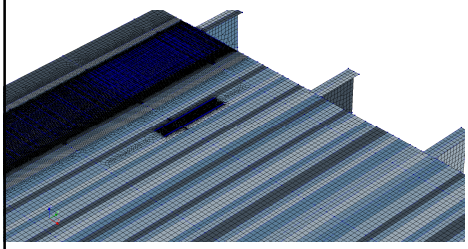
LOADCASES AND DATA POINTS

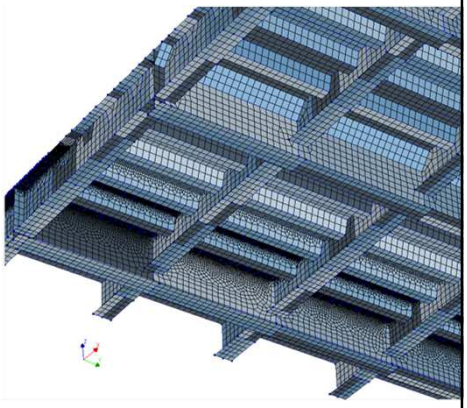




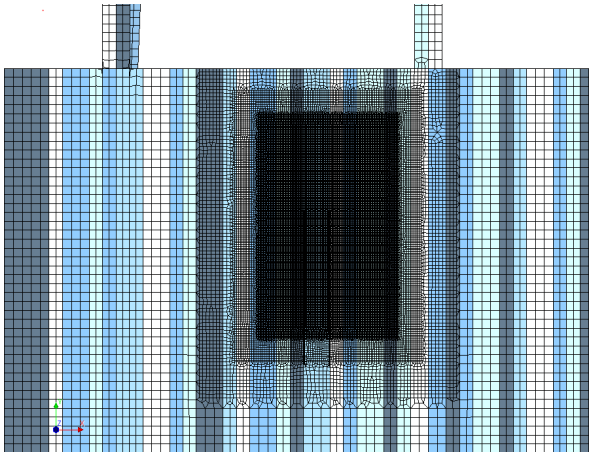
MESH

- Locally refined near hotspots and hotspots locations



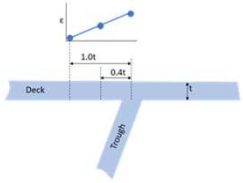


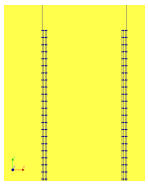
MESH (2)

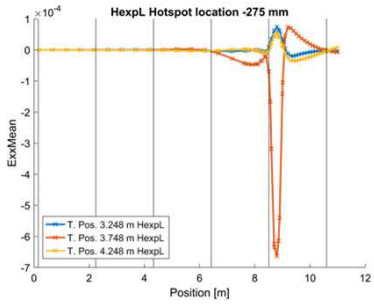


RESULTS

- Strain in transversal direction
  - Hotspot strain by extrapolating strain at 0.4T and 1.0T
- Influence lines in both transversal and longitudinal direction
  - Example longitudinal direction
  - Three different transversal positions







PARAMETRIC MODELLING

- Goal:
  - Automated modelling of different orthotropic bridges based on a general input file
- Both pre and post processing is automated
- Input is based on simple input file
  - Multiple groups of parameters
  - Automatically extended by adding a parameter

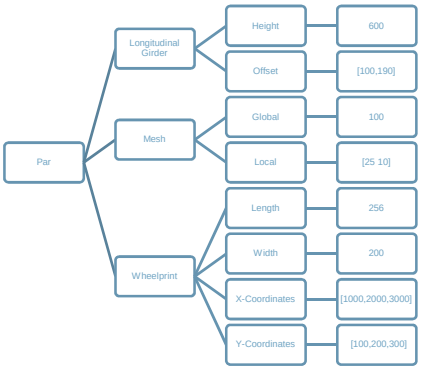
```
# Create by Sjoerd Hengeveld, TNO 22-09-2017
[trough]
 nr = [4,4,4,4,2]i                    # Number of throught per segment (so between longitudinal girders
 x0 = [598,158,358,358,158]           #distance from girder to center of first trough (to position all
 c2c = 600                            # c2c distance
 height=223                           # height
 widthTop = 380                       #thickness
 widthBottom =170                     #width throught bottom side
 thickness = 6

[girder1]
 c2c= [156,2066,2090,2090,2090,2090,715]   # c2c distance
 height=591   # height of longitudinal girder
 thicknessFlange=25  #thicknessbottomflagne
 thicknessWeb =12      #thicknessWeb
 width=380     # width bottomflange
 bModelInTrough = 1

[deck]
 thickness= 12

[wheel]
 load = -1              # uniformly distributed load of wheel print
 nrX  = 11                    # number of wheel prints in transversal (x) direction
 nrY  = 11          # number of wheel prints in longitudinal (y) direction
 length = 356       # length wheel print type A
 width  = 186       # width wheel print type A
 xi0 = 0001
 yi0 = 0001
 type = 2
 locX = [3445]
 locY = [6832]

[mesh]
 seedWP = 10
 seedReg = 100
 seedTransition = [10,20]
nrElemTransition = 101         # number of elements per bounding box
nrElemChanging = 31         # number of elements per bounding box

[transformations]
 girderInd  = [5,0]i    %For numbering see sketch
 girderYdz  = [-100,-150]
 girderLind  = [0,2,0]i
 girderLdz   =[70,-70,-132] % Transformations are based on top part of flange

[datalines]
 xLoc = [0.4,1.4] # assuming x=thicknesssDecka
 yLoc = [0,1200,-580]  # [nrI,+dy,-dy] distance from girder
 troughNr = 5   %multiple numbers can be applied FIRST TROUGH iS 1
 dx = 25

[general]
 projectFolder = simulations1812
 projectName = TestParametricModel001
 mrLc        = 1
```

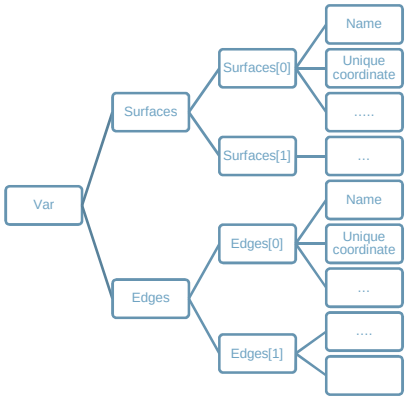

MODULE ORTHODECK

- Top level definitions
  - Modelling is divided in parts
  - Adjustments can be done after every stage
- Common interface: FunctionName(var,globdat)
  - Globdat: global database, a bibliography containing all properties
  - Var: variables, a bibliography containing all runtime variables and state variables
- Only non-automated part is meshing
  - Hard to automate
  - Less prone to errors

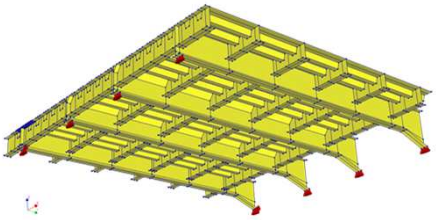
```
#####################################################
####
####  Author: Sjoerd Hengeveld  TNO
####
#####################################################
import os
import ModuleOrthoDeck
import pickle
from ModuleOrthoDeck import *
#Initialize Model
startTimeDef(filePath,'a')
globdat = createGlobalDatabase(filePath,'input.ini')
initializeNewModel(var,{})
var =readVarFromFile(globdat,'init')
#Create Geometry
addTransversalPart(var,globdat)
addDeck(var,globdat)
addLongitudinalGirders(var,globdat)
addGeometryToSurfaces(var,globdat)
addMaterialsToBridge(var,globdat)

#Add loads
addAllWheelPrintsLocation(var,globdat)
addBoundingBox(var,globdat)

#Do adjustment to model
transformations(var,globdat)
addElementGroupToAnalyse(var,globdat)a

#Add supports
addSupports(var,globdat)a

#add analyses and create batchfiles
addAnalysisToBridge(var,globdat)

finishModel(var,globdat,'completeModel')
#####################################################
print('Finished everything')
```

COMMON DEFINITION SIGNATURE





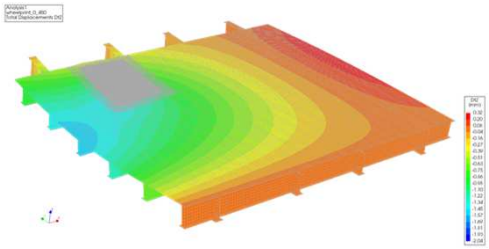
EXAMPLE: BC'S

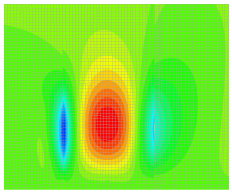


```
def addSupports(var,par):
#Definition which add Z-supports in model
#First select correct surfaces
  addSet(GEOMETRYSUPPORTSET,'BoundaryConditions')
  tempStrZ = 'BC1_Z'
  tempStrY = 'BC1_Y'
  tempStrX = 'BC1_X'

  createPointSupport(tempStrY,'BoundaryConditions')
  createPointSupport(tempStrX,'BoundaryConditions')
  createLineSupport(tempStrZ,'BoundaryConditions')

  setParameter(GEOMETRYSUPPORT,tempStrZ,"AXES",[1,2])
  setParameter(GEOMETRYSUPPORT,tempStrZ,"TRANSL",[0,0,1])
  setParameter(GEOMETRYSUPPORT,tempStrZ,"ROTATI",[0,0,0])

  setParameter(GEOMETRYSUPPORT,tempStrY,"AXES",[1,2])
  setParameter(GEOMETRYSUPPORT,tempStrY,"TRANSL",[0,1,0])
  setParameter(GEOMETRYSUPPORT,tempStrY,"ROTATI",[0,0,0])

  setParameter(GEOMETRYSUPPORT,tempStrX,"AXES",[1,2])
  setParameter(GEOMETRYSUPPORT,tempStrX,"TRANSL",[1,0,0])
  setParameter(GEOMETRYSUPPORT,tempStrX,"ROTATI",[0,0,0])

  temp = findObjectList(var['surface'],'type','girderLflange')
  surfaces = findObjectList(temp,'lt',0)

  for ii in range(0,len(surfaces)):
      surf =surfaces[ii]
      ep = surf['edgePoints']
      point =[(ep[0][0]+ep[1][0])/2,ep[0][1],ep[0][2]]
      attach(GEOMETRYSUPPORT,tempStr2,surf['name'],[point])

  attach(GEOMETRYSUPPORT,tempStrX,surfaces[0]['name'],[surfaces[0]['edgePoints'][0]])

  temp = findObjectList(var['surface'],'type','girderLflange')
  surfaces = findObjectList(temp,'lt',len(par['girderT']['c2c']))
  for ii in range(0,len(surfaces)):
      surf =surfaces[ii]
      ep = surf['edgePoints']
      if len(ep)>0:
          point =[(ep[0][0]+ep[1][0])/2,ep[0][1],ep[0][2]]
          attach(GEOMETRYSUPPORT,tempStr2,surf['name'],[point])

  attach(GEOMETRYSUPPORT,tempStrX,surfaces[0]['name'],[surfaces[0]['edgePoints'][0]])
  attach(GEOMETRYSUPPORT,tempStrY,surfaces[0]['name'],[surfaces[0]['edgePoints'][0]])
```

POSTPROCESSING

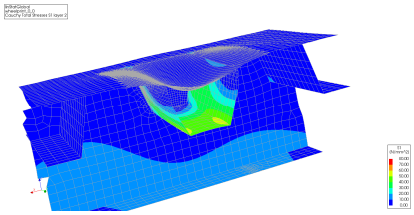
- Per loadcase a single Tabulated output file is generated
- Per set of loadcases a .Dnb file is used to check the global behaviour
- Tabulated outputfiles are postprocessed using MATLAB

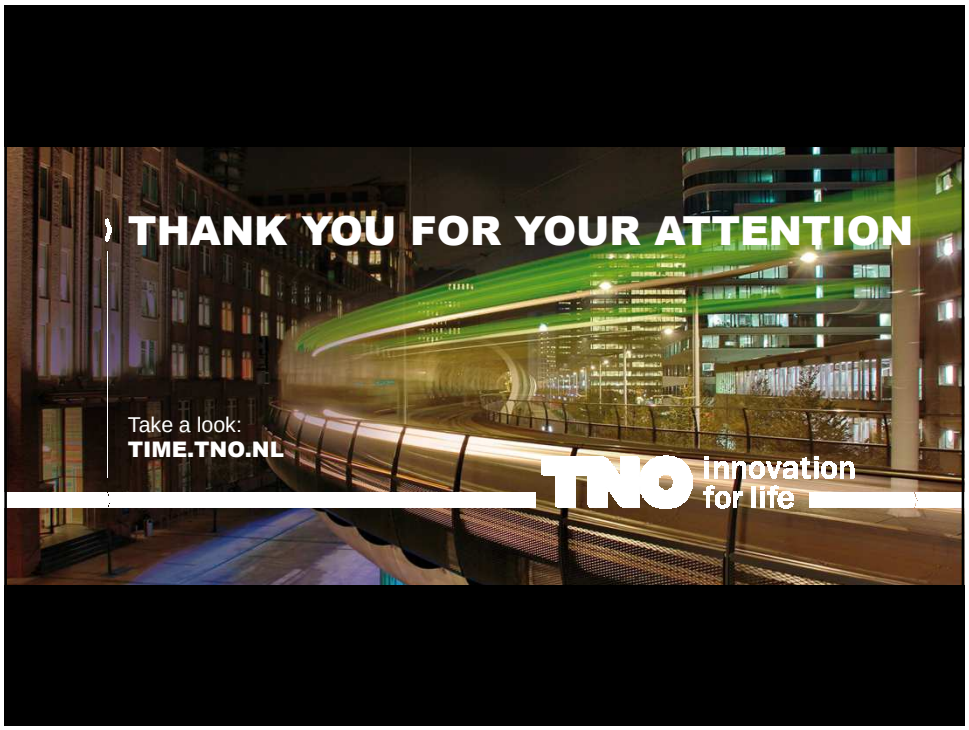




CONCLUSIONS, CHALLENGES & LEARNING POINTS

- For large (parametric) studies repetitive work can be easily reduced by using python interface
- If prefered meshing can be automated, but it is a challenge to add mesh refinements
- Accurate bookkeeping of geometry is needed




THANK YOU FOR YOUR ATTENTION
Take a look:
TIME.TNO.NL
TNO innovation for life