Let $\Omega \subset \mathbb{R}^3$ be a 3d tuning fork composed of rods of diameter r. The geometry is described in Figure 1 and the meshes can be obtained by using `gmsh` (version $>=$ 3.0.0) with the file `tuning_fork_02.geo`.

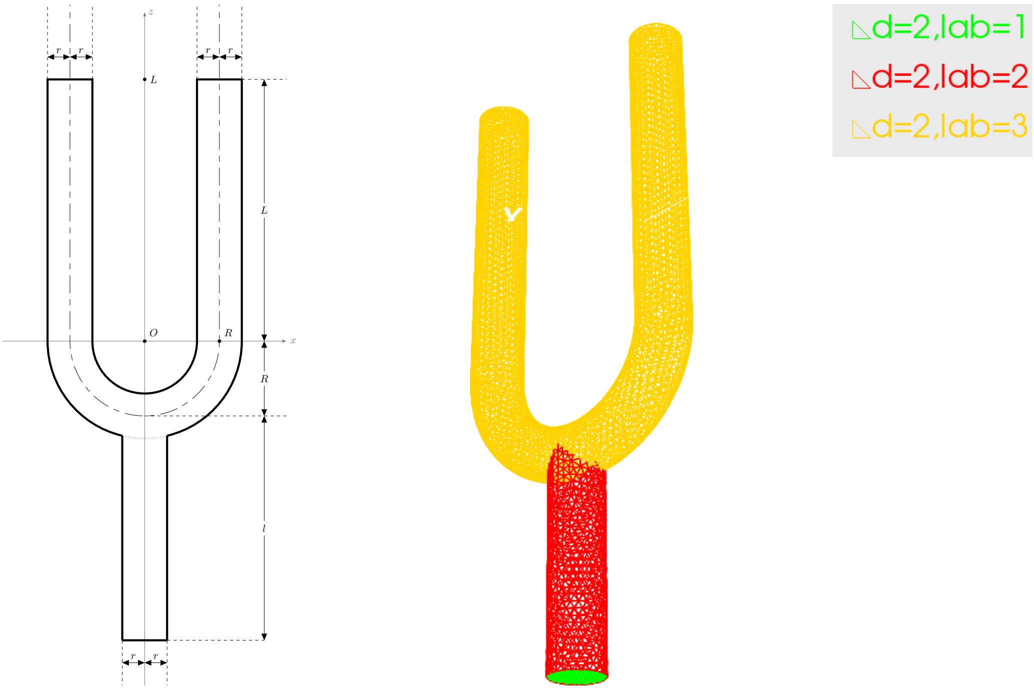

Figure 1: 3d tuning fork geometrical parameters (left) and its boundaries (right)

The eigenvalue problem to solve is the following

Usual EBVP 1 : 3D tuning fork with Dirichlet boundary condition on Γ_1

Find $(\kappa, \boldsymbol{u}) = \mathbb{K} \times \mathrm{H}^1(\Omega)^3$ such that

$$\begin{aligned} -\operatorname{div}(\boldsymbol{\sigma}(\boldsymbol{u})) &= \kappa\rho\boldsymbol{u}, \quad \text{in } \Omega \\ \boldsymbol{\sigma}(\boldsymbol{u}).\boldsymbol{n} &= \mathbf{0} \quad \text{on } \Gamma_2 \cup \Gamma_3, \\ \boldsymbol{u} &= \mathbf{0} \quad \text{on } \Gamma_1. \end{aligned}$$

The geometrical parameters, in millimeters, are $L = 120$, $R = 40$, $l = 100$ and $r = 10$. The isotropic material is made of aluminium and so its Poisson's ratio ν is 0.334, its Young's modulus E is 7.10×10^7 kg.s^{-2}.mm^{-1} and its density ρ is 2.77×10^{-6} kg.mm^{-3}.

In Listing 1 we give parts of Python code to solve this eigenvalue BVP.

```
import numpy as np
import os
from fc_oogmsh import gmsh
from fc_simesh.siMesh import siMesh
from fc_vfemp1.operators import Hmass, StiffElasHoperator
from fc_vfemp1.BVP import BVP,PDE
import fc_vfemp1_eigs.lib as elib
from fc_vfemp1_eigs.sys import get_geo
# Geometrical/Mesh properties:
N=75
L=120 # in mm
r=10  # in mm
R=40  # in mm
l=100 # in mm
# Material properties:
nu=0.334      # Poisson's ratio
E= 71e6       # Modulus of elasticity in kg/s^2/mm ( GPa=10^9 Pa = 10^6 kg/s^2/mm )
rho=2.770e-6 # Material density in kg/mm^3
```

```
# Select Dirichlet boundary:
dirlab=[1] # Dirichlet on Boundary \Gamma_1
# Parameters for eBVPsolve function
NumEigs=9
sigma=0
# Find .geo file
geofile='tuning_fork_02'
(geodir,geofile)=get_geo(3,3,geofile)
geoFile=geodir+os.sep+geofile+'.geo'
options='-smooth 4 -setnumber L %g -setnumber l %g -setnumber r %g -setnumber R 
    %g'%(L,l,r,R)
meshfile=gmsh.buildmesh3d(geoFile,N, force=True,verbose=3,options=options)
Th=siMesh(meshfile)
# Setting the eBVP
mu= E/(2*(1+nu))
lam = E*nu/((1+nu)*(1-2*nu))
Hop=StiffElasHoperator(3,lam,mu)
pde=PDE(Op=Hop)
bvp=BVP(Th,pde=pde)
for lab in dirlab:
  bvp.setDirichlet(lab,[0.,0.,0.])
Bop=Hmass(Th.dim,Th.dim,a0=rho)
# Solving the eBVP
eigVal,eigVec= elib.eBVPsolve(bvp,RHSop=Bop,k=NumEigs,which='LM',sigma=sigma,
    tol=1e-9,maxiter=Th.nq)
```

Listing 1: part of the file `LinearElasticity_tuning_fork3D02_light.py`

One can also use the `run` function of the `fc_vfemp1_eigs.examples.LinearElasticity_tuning_fork3D02` module to solve this eigenvalue BVP. The twenty-four first eigenfunctions given by the following command are represented in Figure 2.

```
from fc_vfemp1_eigs.examples.LinearElasticity_tuning_fork3D02 import run
res=run(N=200,L=120,R=40,l=100,r=10,nu=0.334,E=71e6,rho=2.77e-6,
    Dirichlet=[1],k=24,colormap='jet')
```

GIT : commit 31902706407cc1d09651a642fbf98141860b2933 Date: Wed Jun 14 17:40:47 2017 +0200

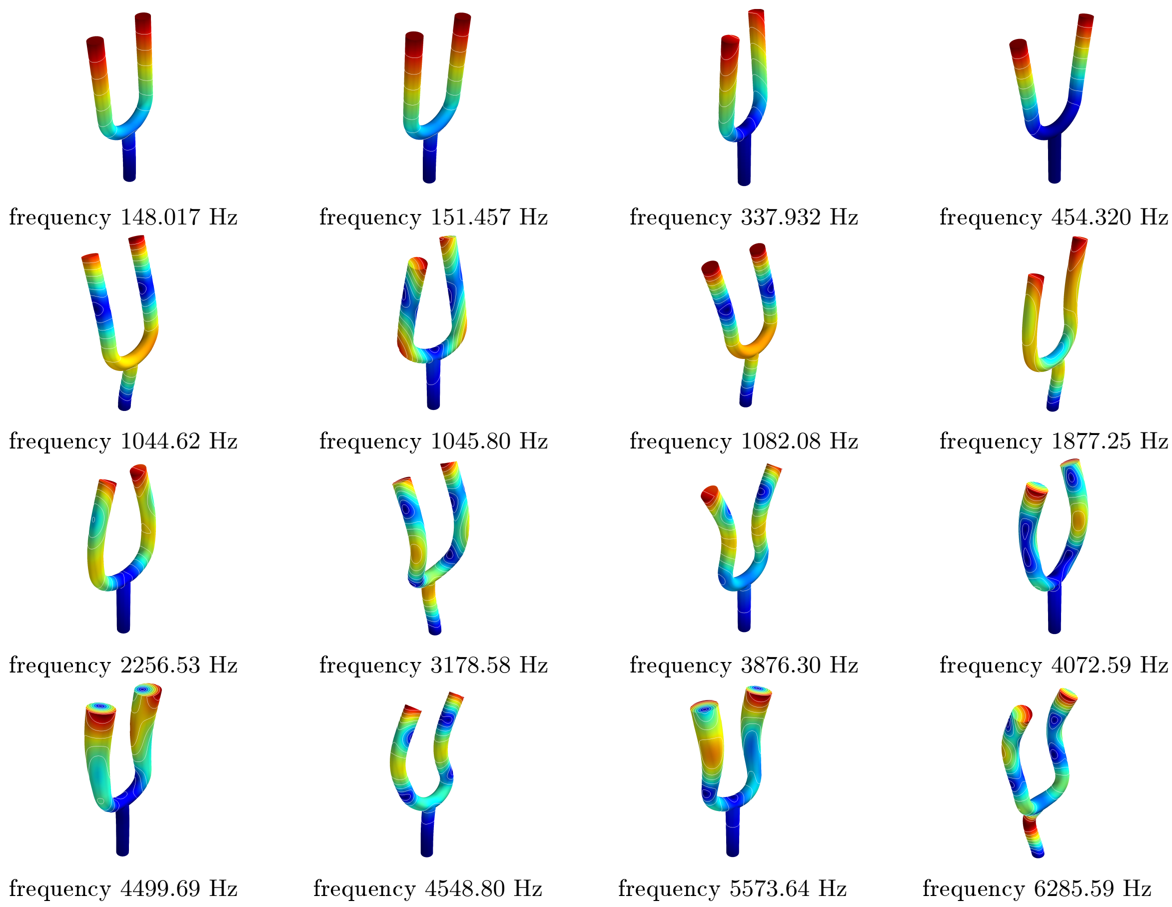

Figure 2: 3D Tuning fork with Neumann boundary conditions on $\Gamma_2 \cup \Gamma_3$ and Dirichlet on Γ_1: eigenvectors of the smallest magnitude eigenvalues streched to a maximum of 10 mm.