Mechanical modal analysis

Contents

1 Optimizing the length of a tuning fork

To illustrate how to use FeenoX in an optimization loop, let us consider the problem of finding the length ℓ_1 of a tuning fork (fig. 1) such that the fundamental frequency on a free-free oscillation is equal to the base A frequency at 440 Hz.
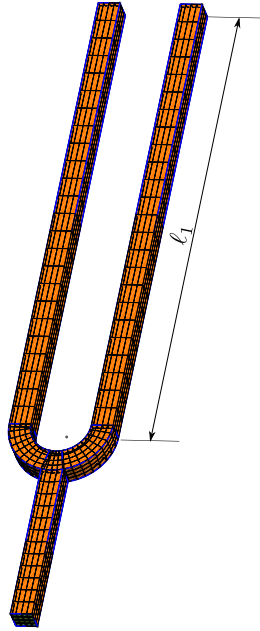

Figure 1: What length ℓ_1 is needed so the fork vibrates at 440 Hz?

The FeenoX input is extremely simple input file, since it has to solve the free-free mechanical modal problem (i.e. without any Dirichlet boundary condition). All it has to do is to print the fundamental frequency.

To find the length ℓ_1, FeenoX is sucessively called from a Python driving script called `fork.py`. This script uses Gmsh's Python API to create the CAD and the mesh of the tuning fork given the geometrical arguments r, w, ℓ_1 and ℓ_2. The parameter n controls the number of elements through the fork's thickness. Here is the driving script without the CAD & mesh details (the full implementation of the function is available in the examples directory of the FeenoX distribution):

```python
import math
import gmsh
import subprocess  # to call FeenoX and read back

def create_mesh(r, w, l1, l2, n):
  gmsh.initialize()
  ...
  gmsh.write("fork.msh")
  gmsh.finalize()
  return len(nodes)

def main():
  target = 440    # target frequency
  eps = 1e-2      # tolerance
```

```python
r = 4.2e-3      # geometric parameters
w = 3e-3
l1 = 30e-3
l2 = 60e-3

for n in range(1,7):   # mesh refinement level
  l1 = 60e-3                # restart l1 & error
  error = 60
  while abs(error) > eps:   # loop
    l1 = l1 - 1e-4*error
    # mesh with Gmsh Python API
    nodes = create_mesh(r, w, l1, l2, n)
    # call FeenoX and read scalar back
    # TODO: FeenoX Python API (like Gmsh)
    result = subprocess.run(['feenox', 'fork.fee'], stdout=subprocess.PIPE)
    freq = float(result.stdout.decode('utf-8'))
    error = target - freq

  print(nodes, l1, freq)
```

Note that in this particular case, the FeenoX input files does not expand any command-line argument. The trick is that the mesh file `fork.msh` is overwritten in each call of the optimization loop. The detailed steps between `gmsh.initialize()` and `gmsh.finalize()` are not shown here,

Since the computed frequency depends both on the length ℓ_1 and on the mesh refinement level n, there are actually two nested loops: one parametric over $n = 1, 2 \dots, 7$ and the optimization loop itself that tries to find ℓ_1 so as to obtain a frequency equal to 440 Hz within 0.01% of error.

```
PROBLEM modal 3D MODES 1   # only one mode needed
READ_MESH fork.msh  # in [m]
E = 2.07e11         # in [Pa]
nu = 0.33
rho = 7829          # in [kg/m^2]

# no BCs! It is a free-free vibration problem
SOLVE_PROBLEM

# write back the fundamental frequency to stdout
PRINT f(1)
```

```
$ python fork.py > fork.dat
$ pyxplot fork.ppl
$
```

2   Five natural modes of a cantilevered wire

Back in College, we had this subject Experimental Physics 101. I had to measure the natural modes of two cantilevered wires and determine the Young modulus of of those measurements. The report is here. Two comments:

1. It is in Spanish
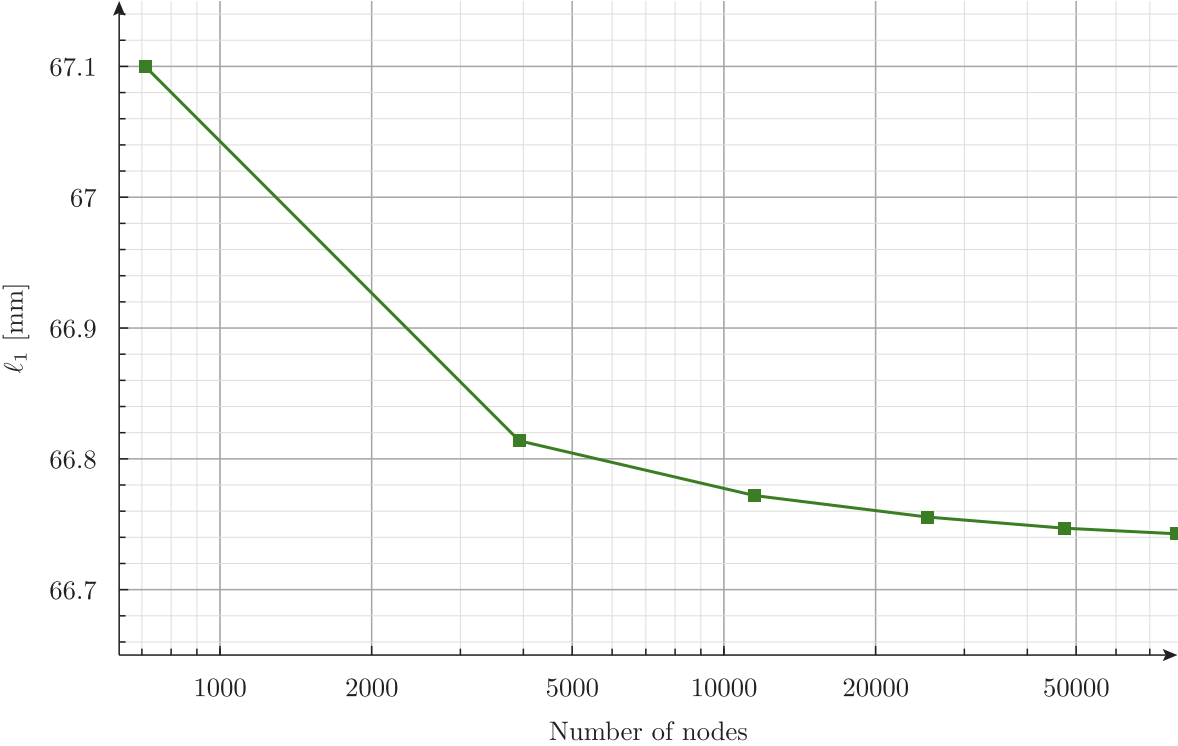2. There was a systematic error and a factor of two sneaked in into the measured values

Figure 2: Estimated length ℓ_1 needed to get 440 Hz for different mesh refinement levels n

Here is a finite-element version of the experimental setup with a comparison to then theoretical values written directly as Markdown tables. The material (either aluminum or copper) and the mesh type (either tet or hex) and be chosen at runtime through command line arguments.

```
DEFAULT_ARGUMENT_VALUE 1 hex        # mesh, either hex or unstruct
DEFAULT_ARGUMENT_VALUE 2 copper     # material, either copper or aluminum

l = 0.5*303e-3    # cantilever wire length [ m ]
d = 1.948e-3      # wire diameter [ m ]


# material properties for copper
m_copper = 0.5*8.02e-3  # total mass (half the measured because of the experimental disposition) [ kg ]
E_copper = 2*66.2e9     # [ Pa ] Young modulus (twice because the factor-two error)

# material properties for aluminum
m_aluminum = 0.5*2.67e-3
E_aluminum = 2*40.2e9

# 'problems properties
E = E_$2                    # [ MPa ]
rho = m_$2/(pi*(0.5*d)^2*l)  # [ kg / m^3 ] density = mass (measured) / volume
nu = 0                      # 'Poissons ratio (does not appear in Euler-Bernoulli)

# analytical solution
VECTOR kl[5]
VECTOR f_euler[5]

# first compute the first five roots ok cosh(kl)*cos(kl)+1
kl[i] = root(cosh(t)*cos(t)+1, t, 3*i-2,3*i+1)

# then compute the frequencies according to Euler-Bernoulli
# note that we need to use SI inside the square root
A = pi * (d/2)^2
I = pi/4 * (d/2)^4
f_euler[i] = 1/(2*pi) * kl(i)^2 * sqrt((E * I)/(rho * A * l^4))

# now compute the modes numerically with FEM
# note that each mode is duplicated as it is degenerated
READ_MESH wire-$1.msh DIMENSIONS 3
PROBLEM modal MODES 10
BC fixed fixed
SOLVE_PROBLEM

# github-formatted markdown table
# compare the frequencies
PRINT "  \$n\$   |    FEM [Hz]   |   Euler [Hz]  |  Relative difference [%]"
PRINT    ":------:|:-------------:|:-------------:|:--------------:"
PRINT_VECTOR SEP "\t|\t" i  %.4g f(2*i-1) f_euler   %.2f 100*(f_euler(i)-f(2*i-1))/f_euler(i)
PRINT
PRINT ": $2 wire over $1 mesh"

# commonmark table
PRINT
PRINT "  \$n\$   |          \$L\$          |        \$\\Gamma\$        |      \$\\mu\$     |        \$M\$"
PRINT    ":------:+:---------------------:+:---------------------:+:-------------:+:--------------:"
PRINT_VECTOR SEP "\t|\t" i "%+.1e" L Gamma "%.4f" mu Mu
PRINT
PRINT ": $2 wire over $1 mesh, participation and excitation factors \$L\$ and \$\\Gamma\$, effective  ↩
```

```
     per-mode and cummulative mass fractions \$\\mu\$ and \$M\$"

# write the modes into a vtk file
WRITE_MESH wire-$1-$2.vtk \
 VECTOR u1 v1 w1 VECTOR u2 v2 w2 VECTOR u3 v3 w3 \
 VECTOR u4 v4 w4 VECTOR u5 v5 w5 VECTOR u6 v6 w6 \
 VECTOR u7 v7 w7 VECTOR u8 v8 w8 VECTOR u9 v9 w9 VECTOR u10 v10 w10

# and into a msh file
WRITE_MESH wire-$1-$2.msh {
 u1 v1 w1
 u2 v2 w2
 u3 v3 w3
 u4 v4 w4
 u5 v5 w5
 u6 v6 w6
 u7 v7 w7
 u8 v8 w8
 u9 v9 w9
 u10 v10 w10
}
```

```
$ gmsh -3 wire-hex.geo
[...]
Info    : Done meshing order 2 (Wall 0.0169025s, CPU 0.016804s)
Info    : 8398 nodes 4676 elements
Info    : Writing 'wire-hex.msh'...
Info    : Done writing 'wire-hex.msh'
Info    : Stopped on Fri Dec 24 17:07:19 2021 (From start: Wall 0.0464517s, CPU 0.133498s)
$ gmsh -3 wire-tet.geo
[...]
Info    : Done optimizing mesh (Wall 0.0229018s, CPU 0.022892s)
Info    : 16579 nodes 13610 elements
Info    : Writing 'wire-tet.msh'...
Info    : Done writing 'wire-tet.msh'
Info    : Stopped on Fri Dec 24 17:07:59 2021 (From start: Wall 2.5798s, CPU 2.64745s)
$ feenox wire.fee
```

n	FEM [Hz]	Euler [Hz]	Relative difference [%]
1	45.84	45.84	0.02
2	287.1	287.3	0.06
3	803.4	804.5	0.13
4	1573	1576	0.24
5	2596	2606	0.38

```
: copper wire over hex mesh
```

n	L	Γ	μ	M
1	+1.3e-03	+4.2e-01	0.1371	0.1371
2	-1.8e-03	-5.9e-01	0.2716	0.4087
3	+9.1e-05	+1.7e-02	0.0004	0.4091
4	-1.7e-03	-3.0e-01	0.1252	0.5343
5	-3.3e-05	-5.9e-03	0.0000	0.5343
6	-9.9e-04	-1.8e-01	0.0431	0.5775

```
7        |        +7.3e-04        |        +1.2e-01        |        0.0221  |        0.5995
8        |        +4.5e-06        |        +7.5e-04        |        0.0000  |        0.5995
9        |        +5.4e-04        |        +9.9e-02        |        0.0134  |        0.6129
10       |        +2.7e-05        |        +4.9e-03        |        0.0000  |        0.6129


: copper wire over hex mesh, participation and excitation factors $L$ and $\Gamma$, effective per-mode and  ↩
    cummulative mass fractions $\mu$ and $M$
$ feenox wire.fee hex copper   | pandoc -o wire-hex-copper.md
$ feenox wire.fee tet copper   | pandoc -o wire-tet-copper.md
$ feenox wire.fee hex aluminum | pandoc -o wire-hex-aluminum.md
$ feenox wire.fee tet aluminum | pandoc -o wire-tet-aluminum.md
```

Table 1: copper wire over hex mesh

n	FEM [Hz]	Euler [Hz]	Relative difference [%]
1	45.84	45.84	0.02
2	287.1	287.3	0.06
3	803.4	804.5	0.13
4	1573	1576	0.24
5	2596	2606	0.38

Table 2: copper wire over hex mesh, participation and excitation factors L and Γ, effective per-mode and cummulative mass fractions μ and M

n	L	Γ	μ	M
1	-1.8e-03	-5.9e-01	0.2713	0.2713
2	+1.3e-03	+4.2e-01	0.1374	0.4087
3	+9.7e-05	+1.8e-02	0.0004	0.4091
4	-1.6e-03	-3.1e-01	0.1251	0.5343
5	-3.5e-05	-6.3e-03	0.0001	0.5343
6	-9.9e-04	-1.8e-01	0.0431	0.5774
7	+7.2e-04	+1.2e-01	0.0221	0.5995
8	-8.6e-06	-1.5e-03	0.0000	0.5995
9	-2.6e-05	-4.7e-03	0.0000	0.5996
10	+5.4e-04	+9.9e-02	0.0134	0.6130

Table 3: copper wire over tet mesh

n	FEM [Hz]	Euler [Hz]	Relative difference [%]
1	45.84	45.84	0.00
2	287.2	287.3	0.05
3	803.4	804.5	0.13

n	FEM [Hz]	Euler [Hz]	Relative difference [%]
4	1573	1576	0.24
5	2596	2606	0.38

Table 4: copper wire over tet mesh, participation and excitation factors L and Γ, effective per-mode and cummulative mass fractions μ and M

n	L	Γ	μ	M
1	-1.9e-03	-6.1e-01	0.2925	0.2925
2	+1.2e-03	+3.8e-01	0.1163	0.4087
3	-1.0e-03	-3.3e-01	0.0861	0.4948
4	+7.0e-04	+2.3e-01	0.0395	0.5343
5	-6.0e-04	-1.9e-01	0.0292	0.5634
6	+4.2e-04	+1.3e-01	0.0140	0.5774
7	-4.0e-04	-1.3e-01	0.0133	0.5908
8	+3.3e-04	+1.1e-01	0.0087	0.5995
9	+3.5e-04	+1.1e-01	0.0096	0.6091
10	-2.2e-04	-6.9e-02	0.0038	0.6129

Table 5: aluminum wire over hex mesh

n	FEM [Hz]	Euler [Hz]	Relative difference [%]
1	61.91	61.92	0.02
2	387.8	388	0.06
3	1085	1086	0.13
4	2124	2129	0.24
5	3506	3519	0.38

Table 6: aluminum wire over hex mesh, participation and excitation factors L and Γ, effective per-mode and cummulative mass fractions μ and M

n	L	Γ	μ	M
1	-6.9e-04	-6.2e-01	0.3211	0.3211
2	+3.6e-04	+3.3e-01	0.0876	0.4087
3	+4.2e-05	+2.4e-02	0.0008	0.4095
4	-5.4e-04	-3.1e-01	0.1248	0.5343
5	+3.7e-05	+2.3e-02	0.0006	0.5349
6	-3.0e-04	-1.9e-01	0.0425	0.5774
7	+2.4e-04	+1.2e-01	0.0221	0.5995

n	L	Γ	μ	M
8	-3.2e-06	-1.6e-03	0.0000	0.5995
9	+1.8e-04	+9.8e-02	0.0132	0.6127
10	-9.5e-06	-5.2e-03	0.0000	0.6128

Table 7: aluminum wire over tet mesh

n	FEM [Hz]	Euler [Hz]	Relative difference [%]
1	61.91	61.92	0.00
2	387.8	388	0.05
3	1085	1086	0.13
4	2124	2129	0.24
5	3506	3519	0.38

Table 8: aluminum wire over tet mesh, participation and excitation factors L and Γ, effective per-mode and cummulative mass fractions μ and M

n	L	Γ	μ	M
1	-6.4e-04	-6.1e-01	0.2923	0.2923
2	+4.0e-04	+3.8e-01	0.1164	0.4087
3	-3.5e-04	-3.3e-01	0.0861	0.4948
4	+2.3e-04	+2.3e-01	0.0395	0.5343
5	-2.0e-04	-1.9e-01	0.0292	0.5634
6	+1.4e-04	+1.3e-01	0.0140	0.5774
7	-1.3e-04	-1.3e-01	0.0133	0.5908
8	+1.1e-04	+1.1e-01	0.0087	0.5995
9	+1.2e-04	+1.1e-01	0.0096	0.6091
10	-7.3e-05	-6.9e-02	0.0038	0.6129