CC BY NC ND

EFEHR-GEM joint webinar

VMTK - Advanced users

Pavia, 20th April 2023

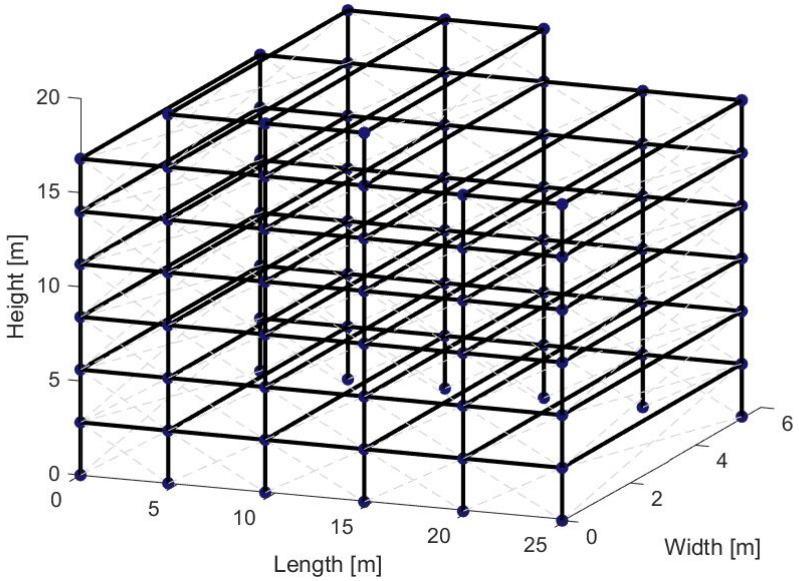
working together
to assess risk

GEM
GLOBAL EARTHQUAKE MODEL

OQ
OPENQUAKE

I have a detailed numerical model can I still use the VMTK?

Using detailed numerical models

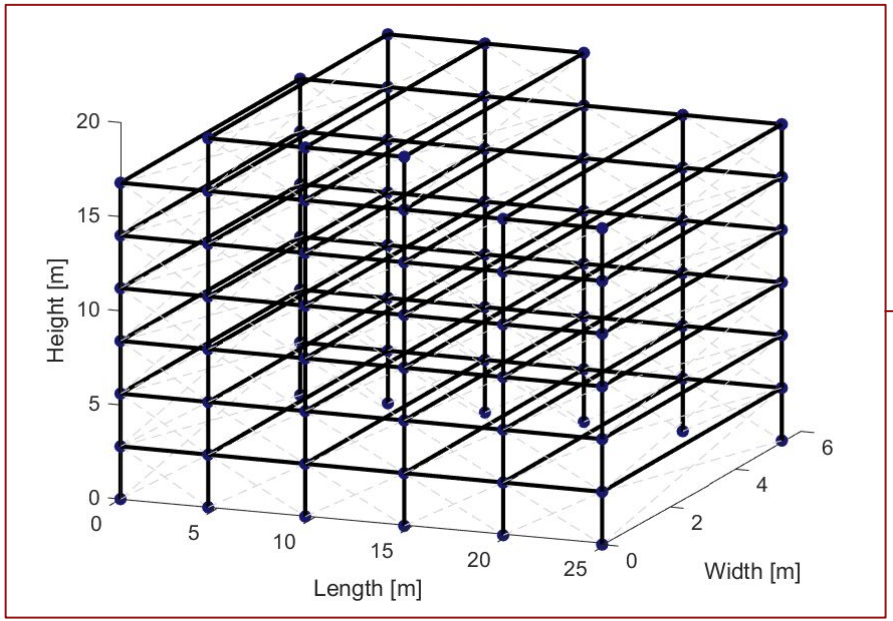


Let's consider this example of a 3D model*:

7 storeys; irregular infilled RC frame; built after 2011 in Colombia

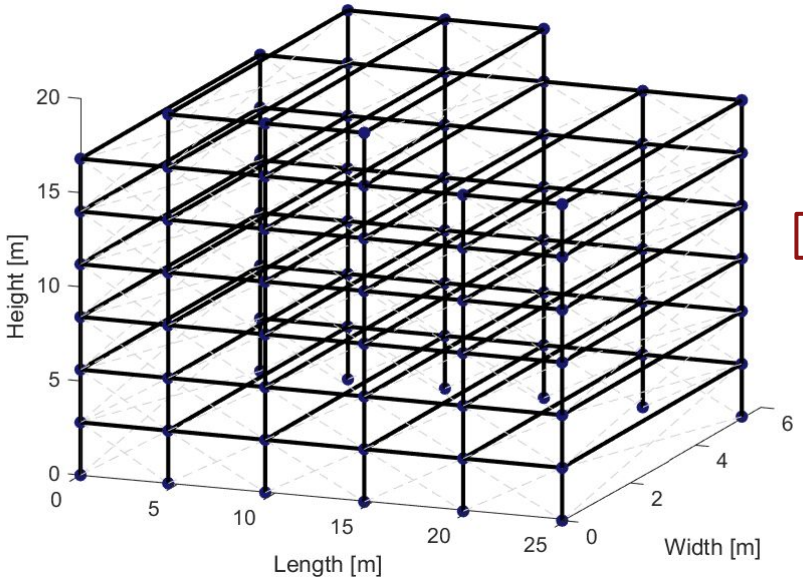
* This numerical model could have been developed in any one of the most popular finite element software packages (e.g. OpenSees, SeismoStruct, SAP2000)

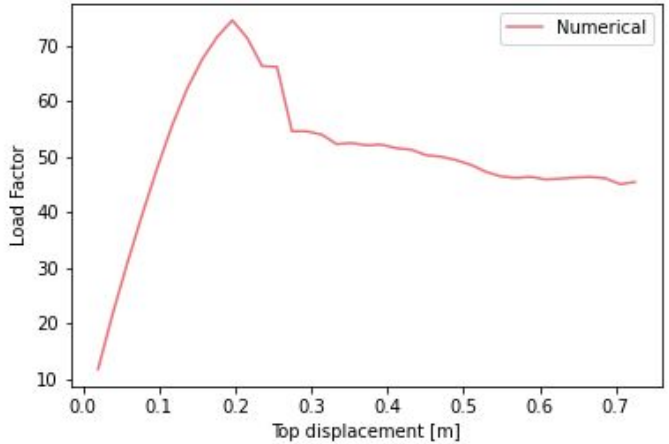
Using detailed numerical models



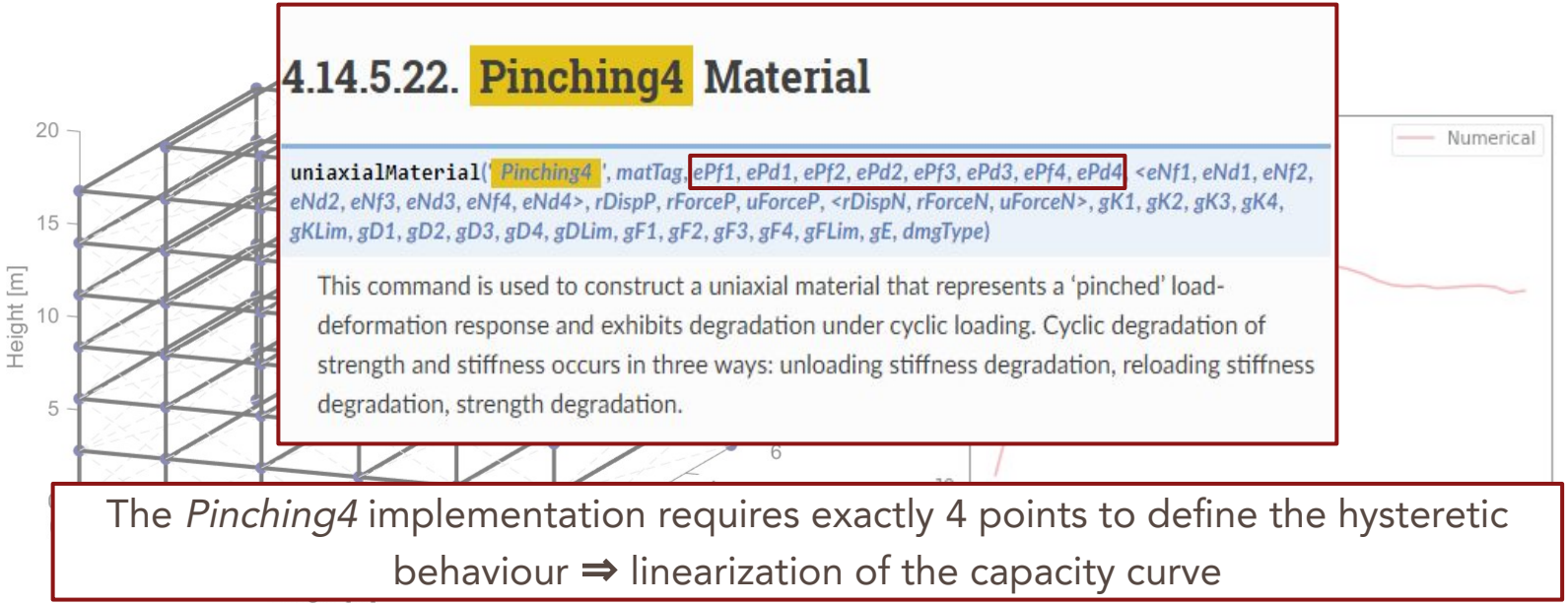
Since performing NLTH analyses on a complex 3D numerical model can be quite demanding. The modeller may opt to use the model simply to estimate the capacity.

Using detailed numerical models





Using detailed numerical models



The *Pinching4* implementation requires exactly 4 points to define the hysteretic behaviour ⇒ linearization of the capacity curve

*https://openseespydoc.readthedocs.io/en/latest/src/Pinching4.html

Using detailed numerical models

master | VMTK-Vulnerability-Modellers-ToolKit / capacity / capacity_start.py | Jump to | Go to file | ...

CatalinaYepes start clean up — Latest commit e1518ef on Feb 26, 2021 — History

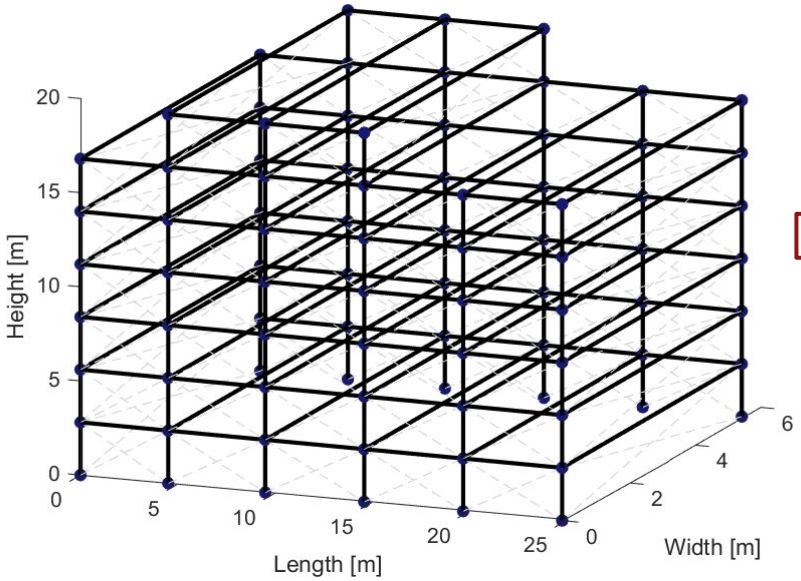
1 contributor

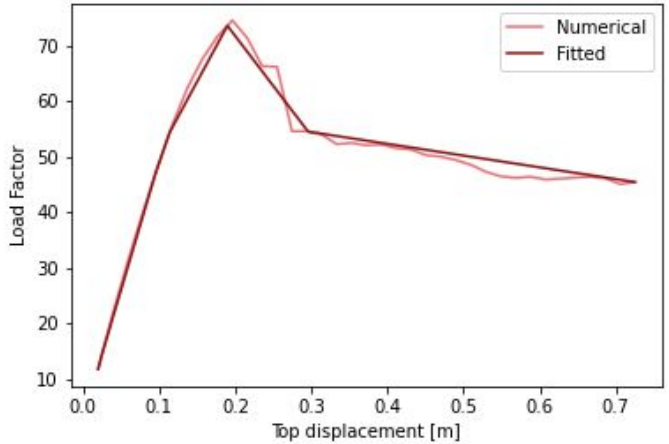
156 lines (104 sloc) | 5.28 KB

```
1   # -*- coding: utf-8 -*-
2   """
3   Created on Thu Dec  5 15:02:10 2019
4
5   @author: LuisMartins
6   """
7   import numpy as np
8   import math
9   import os
10
11  #%%
12  class Capacity:
13
14        # The capacity object is defined by its tag, spectral di
15        # spectral acceleration values
16
17        # tag - string that connects the building class to the capacity
18        # spectral_displ - x values on the capacity object
19        # spectral_accel - y values on the capacity object
20
21        def __init__(self,tag,spectral_displ,spectral_accel,units):
22              self.tag=tag
23              self.spectral_displ=spectral_displ
```

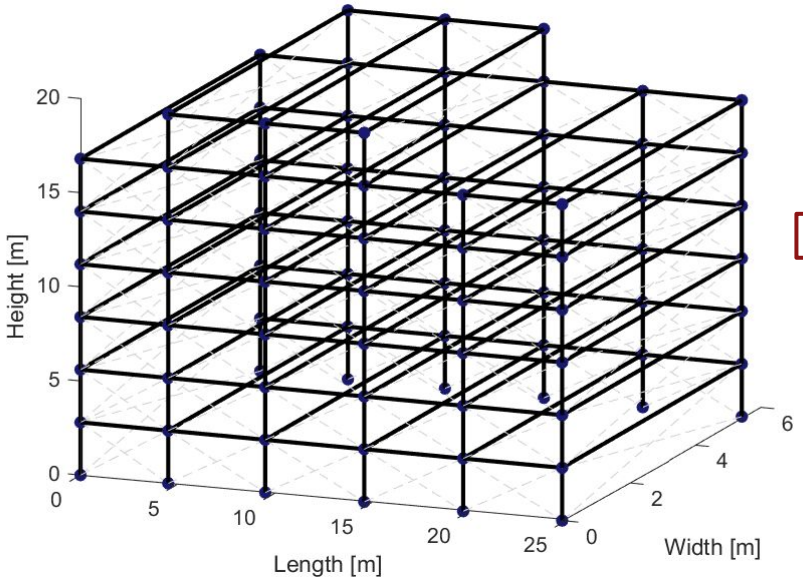
```
def polylinear_curve_fit(x_array,y_array,n_pts=10):

      # This functon takes a generic curve and fits n linear segments through
      # adaptive interpolation

      indices=np.array(np.round(np.linspace(0,np.size(y_array)-1,n_pts,endpoint=True),decimals=0),dtype=int)
      derivative_approx = np.diff(y_array[indices])
      inv_diff=1/np.abs(derivative_approx)
      weight_of_section=inv_diff/np.sum(inv_diff)

      total_range=np.max(x_array)-np.min(x_array)
      size_of_section=weight_of_section*total_range

      # fitted curve points
      x_array_fit=x_array[0]+np.append([0],np.cumsum(size_of_section))
      y_array_fit=np.interp(x_array_fit,x_array,y_array)

      return x_array_fit,y_array_fit
```

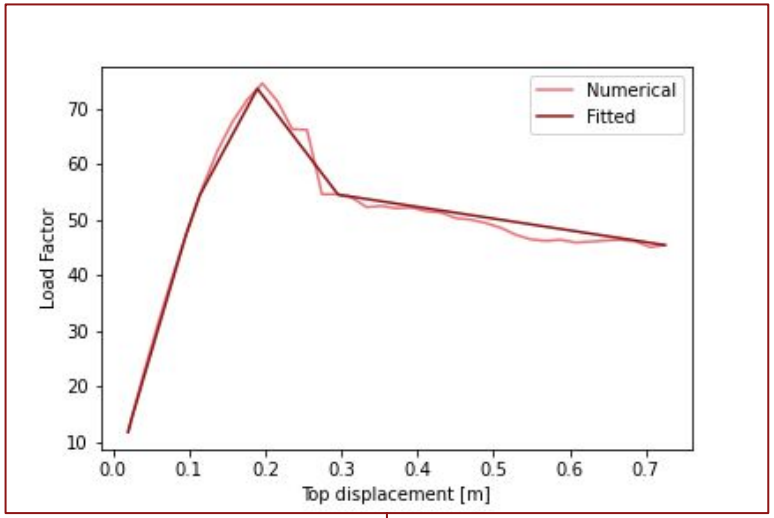
Using detailed numerical models





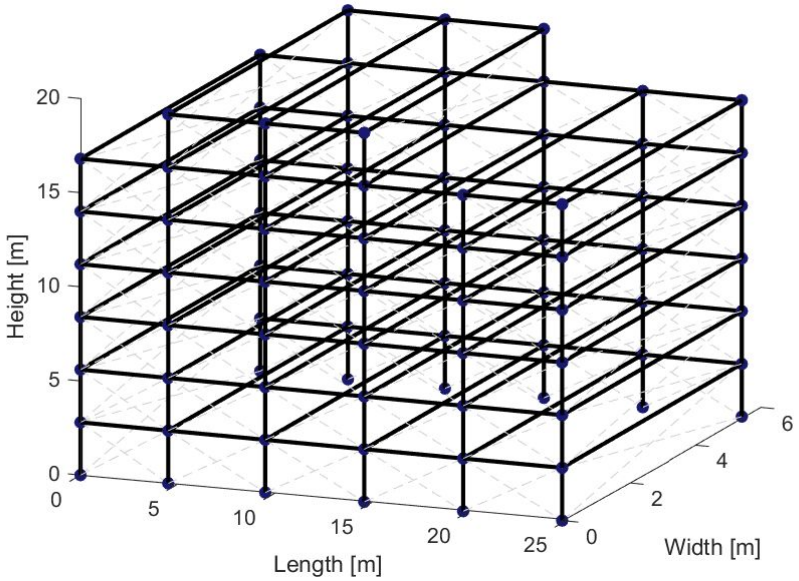
Using detailed numerical models





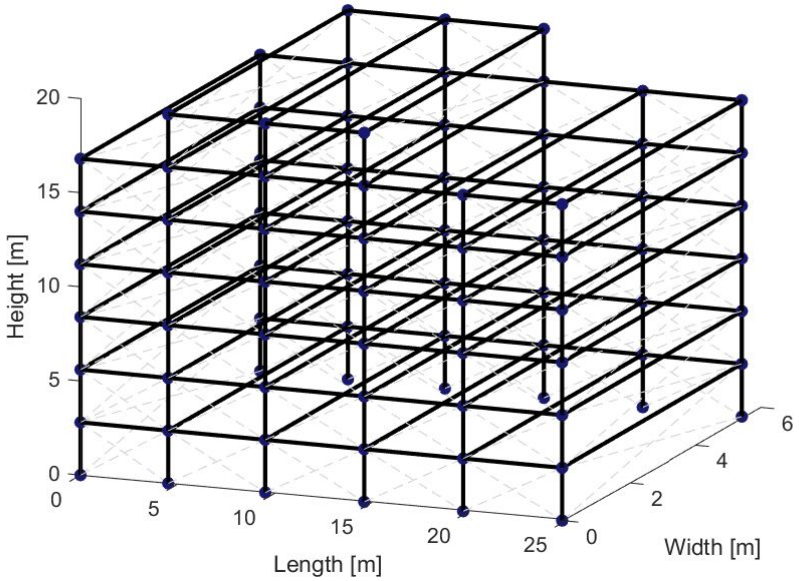
Now the modeller can use this simplified capacity curve in the VMTK (after conversion to ADRS) and proceed with their NLTH analyses

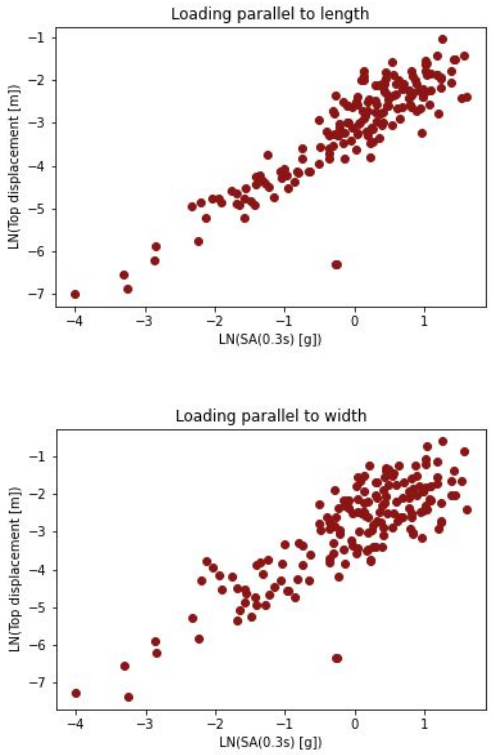
Using detailed numerical models



Let's now imagine that the modeller ran nonlinear time-history analyses with the 3D model and recorded the structural response

Using detailed numerical models





GLOBAL QUAKE MODEL .ORG

Using detailed numerical models

master · VMTK-Vulnerability-Modellers-ToolKit / fragility_vulnerability / fragility_vulnerability_start.py · Jump to · Go to file

lmartins88 changes to xml writer and remove unnecess

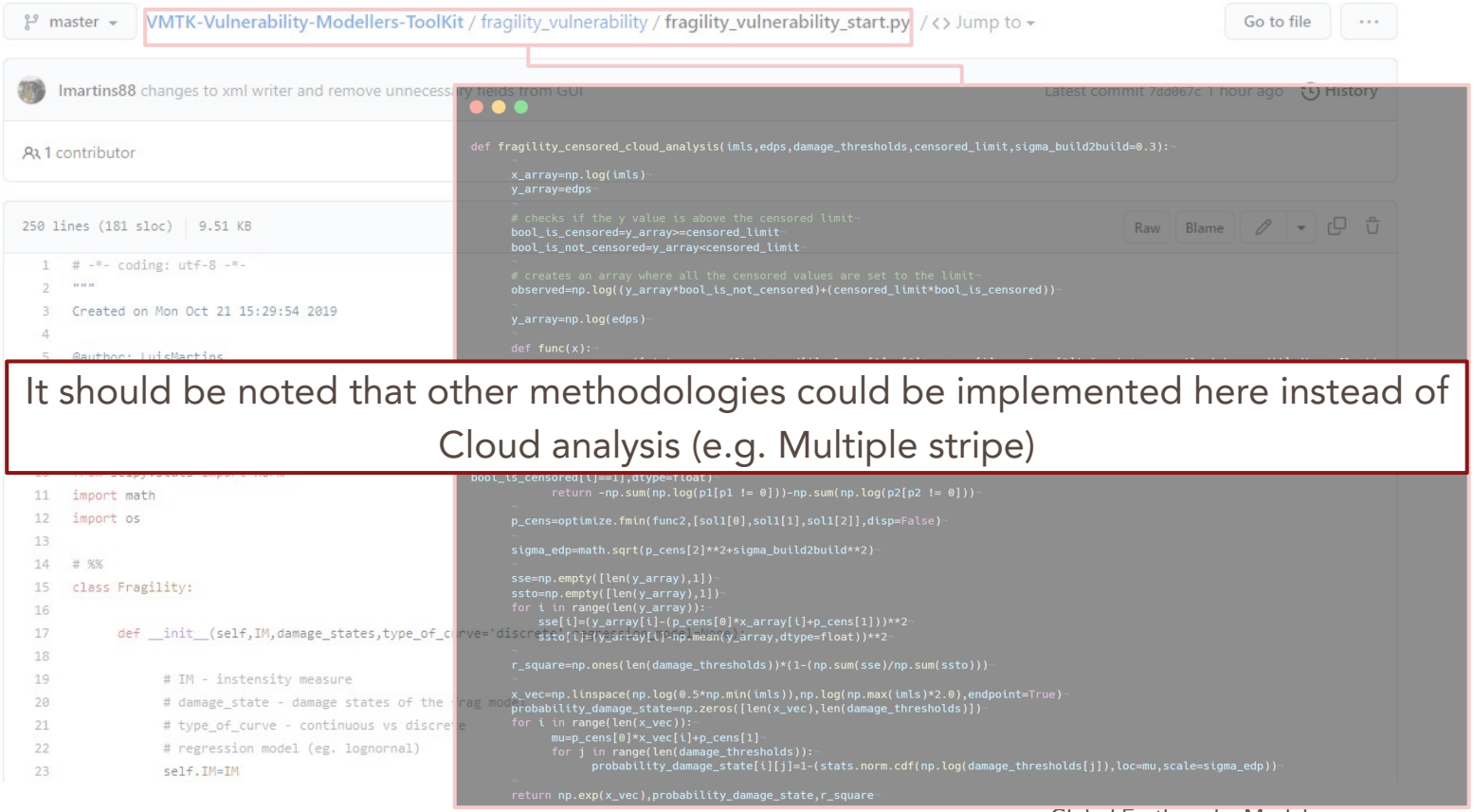
1 contributor

250 lines (181 sloc) | 9.51 KB

```
1   # -*- coding: utf-8 -*-
2   """
3   Created on Mon Oct 21 15:29:54 2019
4
5   @author: LuisMartins
6   """
7
8   import numpy as np
9   from scipy import integrate, optimize, stats
10  from scipy.stats import norm
11  import math
12  import os
13
14  # %%
15  class Fragility:
16
17      def __init__(self,IM,damage_states,type_of_c
18
19          # IM - instensity measure
20          # damage_state - damage states of the
21          # type_of_curve - continuous vs discre
22          # regression model (eg. lognornal)
23          self.IM=IM
```

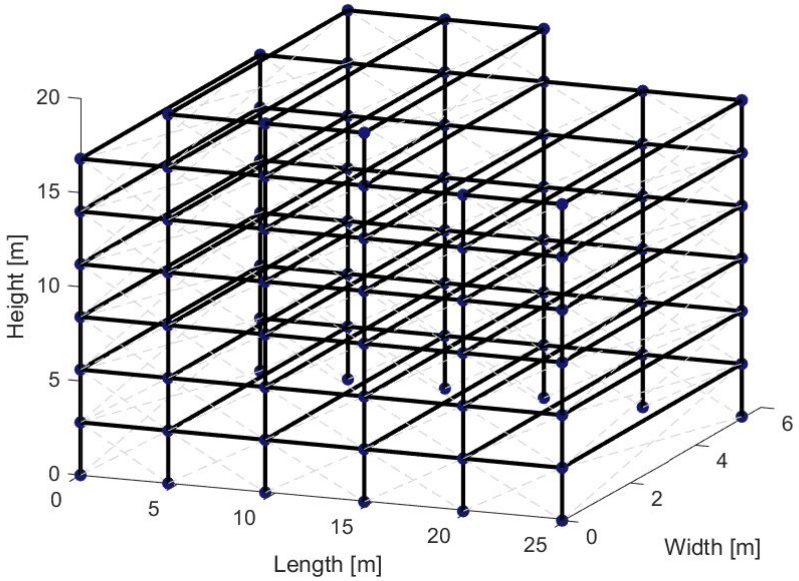
```
def fragility_censored_cloud_analysis(imls,edps,damage_thresholds,censored_limit,sigma_build2build=0.3):

    x_array=np.log(imls)
    y_array=edps

    # checks if the y value is above the censored limit
    bool_is_censored=y_array>=censored_limit
    bool_is_not_censored=y_array<censored_limit

    # creates an array where all the censored values are set to the limit
    observed=np.log((y_array*bool_is_not_censored)+(censored_limit*bool_is_censored))

    y_array=np.log(edps)

    def func(x):
        p = np.array([stats.norm.pdf(observed[i], loc=x[1]+x[0]*x_array[i], scale=x[2]) for i in range(len(observed))],dtype=float)
        return -np.sum(np.log(p[p!= 0]))
    sol1=optimize.fmin(func,[1,1,1],disp=False)

    def func2(x):
        p1 = np.array([stats.norm.pdf(observed[i], loc=x[1]+x[0]*x_array[i], scale=x[2]) for i in range(len(observed)) if
bool_is_censored[i]==0],dtype=float)
        p2 = np.array([1-stats.norm.cdf(observed[i], loc=x[1]+x[0]*x_array[i], scale=x[2]) for i in range(len(observed)) if
bool_is_censored[i]==1],dtype=float)
        return -np.sum(np.log(p1[p1 != 0]))-np.sum(np.log(p2[p2 != 0]))

    p_cens=optimize.fmin(func2,[sol1[0],sol1[1],sol1[2]],disp=False)

    sigma_edp=math.sqrt(p_cens[2]**2+sigma_build2build**2)

    sse=np.empty([len(y_array),1])
    ssto=np.empty([len(y_array),1])
    for i in range(len(y_array)):
        sse[i]=(y_array[i]-(p_cens[0]*x_array[i]+p_cens[1]))**2
        ssto[i]=(y_array[i]-np.mean(y_array,dtype=float))**2

    r_square=np.ones(len(damage_thresholds))*(1-(np.sum(sse)/np.sum(ssto)))

    x_vec=np.linspace(np.log(0.5*np.min(imls)),np.log(np.max(imls)*2.0),endpoint=True)
    probability_damage_state=np.zeros([len(x_vec),len(damage_thresholds)])
    for i in range(len(x_vec)):
        mu=p_cens[0]*x_vec[i]+p_cens[1]
        for j in range(len(damage_thresholds)):
            probability_damage_state[i][j]=1-(stats.norm.cdf(np.log(damage_thresholds[j]),loc=mu,scale=sigma_edp))

    return np.exp(x_vec),probability_damage_state,r_square
```

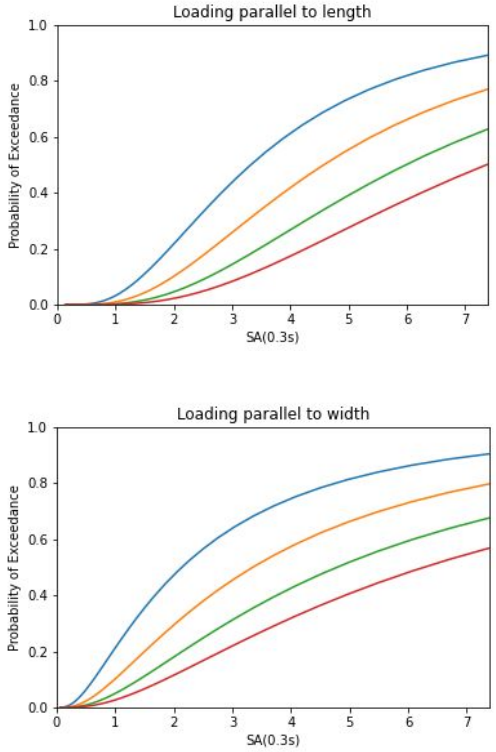
Using detailed numerical models



It should be noted that other methodologies could be implemented here instead of Cloud analysis (e.g. Multiple stripe)

Using detailed numerical models





Using detailed numerical models

master ▾ VMTK-Vulnerability-Modellers-ToolKit / fragility_vulnerability / **fragility_vulnerability_start.py** / <> Jump to ▾ Go to file ···

lmartins88 changes to xml writer and remove unnecessary fields from GUI — Latest commit 7dd067c 1 hour ago — History

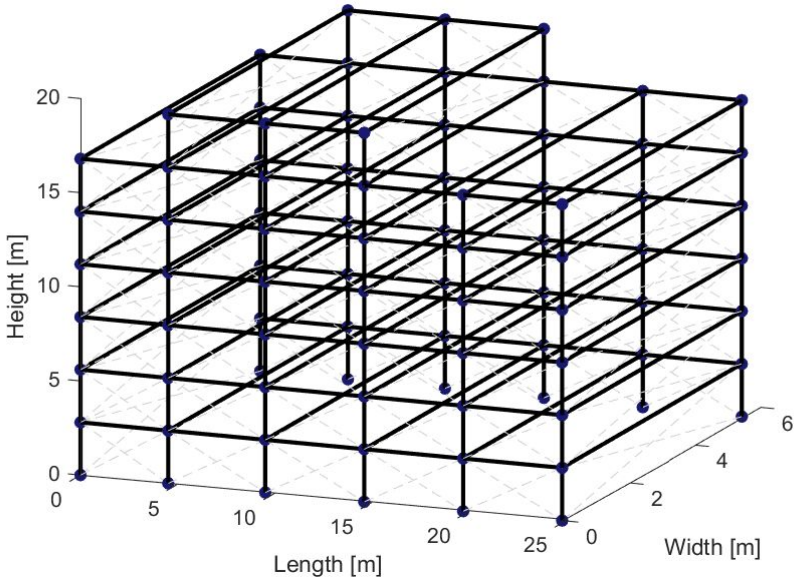
1 contributor

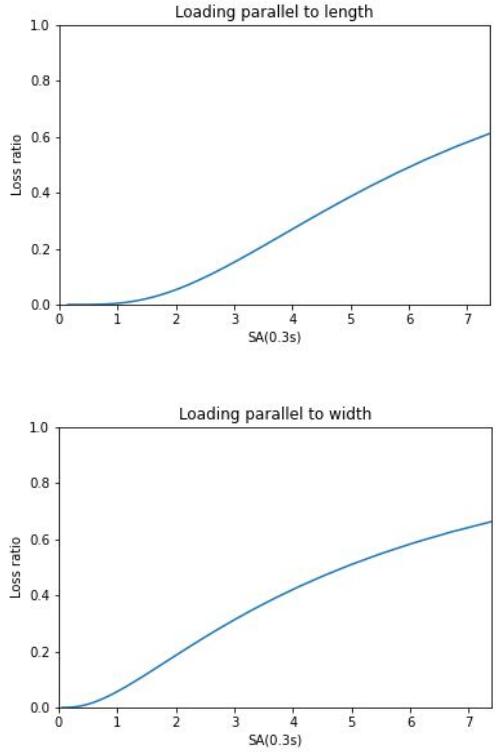
250 lines (181 sloc) | 9.51 KB

```
# -*- coding: utf-8 -*-
"""
Created on Mon Oct 21 15:29:54 2019

@author: LuisMartins
"""

import numpy as np
from scipy import integrate, optimize, stats
from scipy.stats import norm
import math
import os

# %%
class Fragility:

      def __init__(self,IM,damage_states,type_of_curve='discrete',regression_model=None):

            # IM - instensity measure
            # damage_state - damage states of the frag model
            # type_of_curve - continuous vs discrete
            # regression model (eg. lognornal)
            self.IM=IM
```

```
def calculate_vul_curve_dam_to_loss(imls,probability_damage_state,consequence_model):

      if len(consequence_model)!=np.size(probability_damage_state,1):
            raise Exception('Mismatch between the fragility consequence models!')

      if len(imls)!=np.size(probability_damage_state,0):
            raise Exception('Mismatch between the number of IMLs and fragility models!')

      loss_array=np.zeros([len(imls),1])
      for i in range(len(imls)):
            for j in range(0,np.size(probability_damage_state,1)):
                  if j==(np.size(probability_damage_state,1)-1):
                        loss_array[i,0]=loss_array[i,0]+probability_damage_state[i,j]*consequence_model[j]
                  else:
                        loss_array[i,0]=loss_array[i,0]+(probability_damage_state[i,j]-probability_damage_state[i,j+1])*consequence_model[j]

      vul_curve=np.column_stack((imls,loss_array))
      vul_curve_array=vul_curve[vul_curve[:,0].argsort()]

      return vul_curve_array
```

Using detailed numerical models





Can I use a different ground motion selection technique?

Ground motion selection

master ▾ VMTK-Vulnerability-Modellers-ToolKit / demand / select_gmrs_start.py / <> Jump to ▾ Go to file ···

CatalinaYepes start clean up — Latest commit e1518ef on Feb 26, 2021 — History

1 contributor

110 lines (83 sloc) | 3.97 KB — Raw Blame

```
# -*- coding: utf-8 -*-
"""
Created on Thu Jan  9 10:44:18 2020

@author: LuisMartins
"""

import numpy as np
import os
import random
import math

from itertools import repeat
import pyrotd
pyrotd.processes = 1
#%%

def select_from_folder_with_imls(imls,folderpath,ngmrs_per_iml,min_scale_factor,max_scale_factor,period=[0]):

        files=os.listdir(folderpath)
        files_arr=np.array(files)
        imls=np.append(imls,99)

        ims_array=np.zeros([len(files),1])

        for i in range(len(files)):
```

Users are welcomed to add their own selection methods to better suit their analysis requirements (e.g. Multiple stripe)

How is degradation implemented? Can I modify the default modelling parameters?

Hysteretic models

master | VMTK-Vulnerability-Modellers-ToolKit / analysis / nlth_on_sdof.py | Jump to | Go to file

CatalinaYepes start clean up — Latest commit e1518ef on Feb 26, 2021 — History

1 contributor

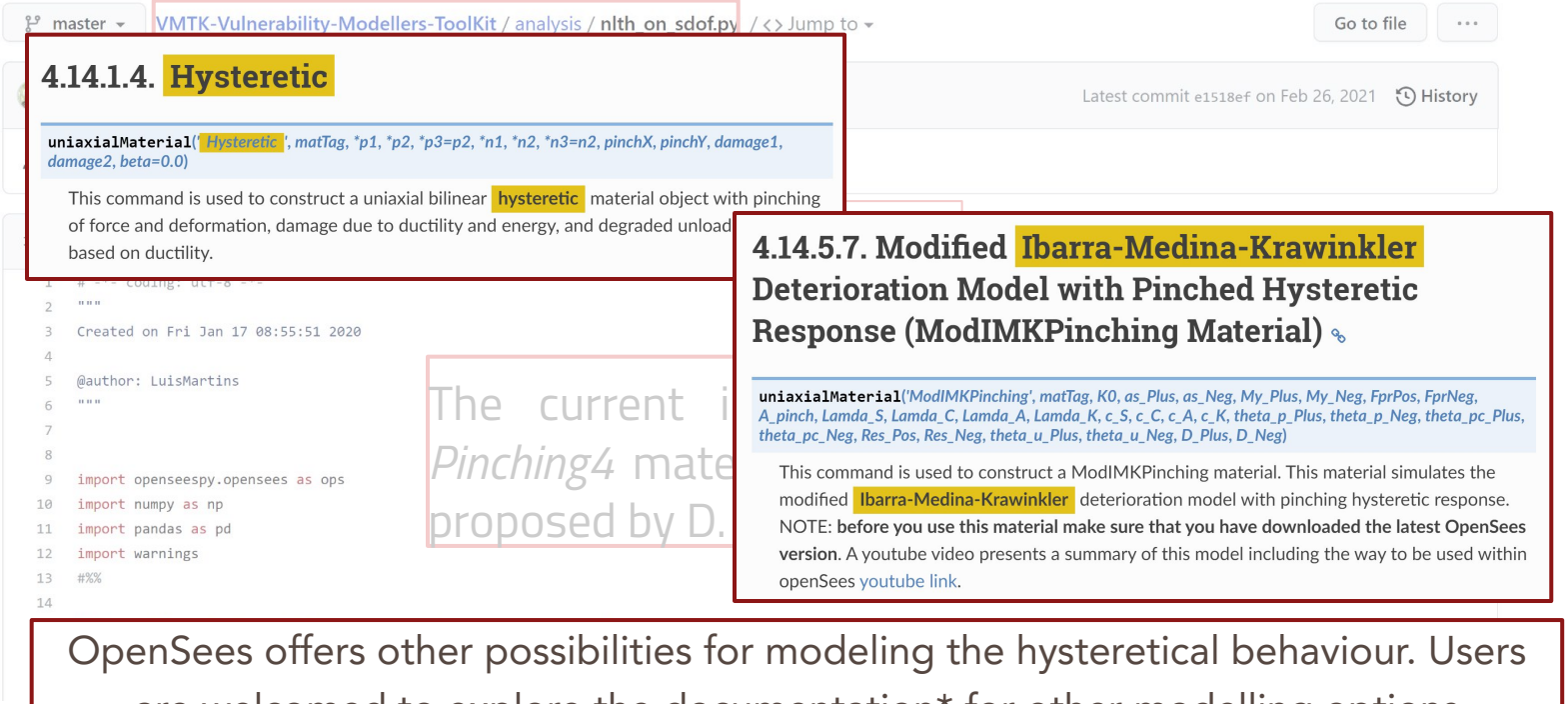
154 lines (122 sloc) | 5.01 KB — Raw | Blame

```
# -*- coding: utf-8 -*-
"""
Created on Fri Jan 17 08:55:51 2020

@author: LuisMartins
"""


import openseespy.opensees as ops
import numpy as np
import pandas as pd
import warnings
#%%

def run_nlth_analysis_on_sdof_ops_py(capacity_curve,gmr,damping,degradation):

      cap_df=pd.DataFrame(capacity_curve)
```

The current implementation makes use of the *Pinching4* material and the degradation parameters proposed by D. Vamvatsikos*

*http://users.ntua.gr/divamva/software.html

Hysteretic models



OpenSees offers other possibilities for modeling the hysteretical behaviour. Users are welcomed to explore the documentation* for other modelling options

*https://openseespydoc.readthedocs.io/en/latest

Hysteretic models

master ▾ VMTK-Vulnerability-Modellers-ToolKit / analysis / nlth_on_sdof.py / <> Jump to ▾ Go to file ···

CatalinaYepes start clean up — Latest commit e1518ef on Feb 26, 2021 — History

1 contributor

154 lines (122 sloc) | 5.01 KB

```
1  # -*- coding: utf-8 -*-
2  """
3  Created on Fri Jan 17 08
4
5  @author: LuisMartins
6  """
7
8
9  import openseespy.opense
10 import numpy as np
11 import pandas as pd
12 import warnings
13 #%%
14
15 def run_nlth_analysis_or
16
17       cap_df=pd.DataFram
```

```
      matTag_pinching=10
      if degradation==True:
            matargs=[f_vec[1,0],d_vec[1,0],f_vec[2,0],d_vec[2,0],f_vec[3,0],d_vec[3,0],f_vec[4,0],d_vec[4,0],

-1*f_vec[1,0],-1*d_vec[1,0],-1*f_vec[2,0],-1*d_vec[2,0],-1*f_vec[3,0],-1*d_vec[3,0],-1*f_vec[4,0],-1*d_vec[4,0],
                                    0.5,0.25,0.05,
                                    0,0.1,0,0,0.2,
                                    0,0.1,0,0,0.2,
                                    0,0.4,0,0.4,0.9,
                                    10,'energy']

      else:
            matargs=[f_vec[1,0],d_vec[1,0],f_vec[2,0],d_vec[2,0],f_vec[3,0],d_vec[3,0],f_vec[4,0],d_vec[4,0],

-1*f_vec[1,0],-1*d_vec[1,0],-1*f_vec[2,0],-1*d_vec[2,0],-1*f_vec[3,0],-1*d_vec[3,0],-1*f_vec[4,0],-1*d_vec[4,0],
                                    0.5,0.25,0.05,
                                    0,0,0,0,0,
                                    0,0,0,0,0,
                                    0,0,0,0,0,
                                    10,'energy']

      ops.uniaxialMaterial('Pinching4', matTag_pinching,*matargs)
      matTag_minmax=int(matTag_pinching/10)
      ops.uniaxialMaterial('MinMax', matTag_minmax,matTag_pinching, '-min',-1*d_vec[4,0], '-max', d_vec[4,0])
```

Damping

master / VMTK-Vulnerability-Modellers-ToolKit / analysis / nlth_on_sdof.py / Jump to — Go to file

CatalinaYepes start clean up — Latest commit e1518ef on Feb 26, 2021 — History

1 contributor

154 lines (122 sloc) | 5.01 KB — Raw | Blame

```
# -*- coding: utf-8 -*-
"""
Created on Fri Jan 17 08:55:51 2020

@author: LuisMartins
"""


import openseespy.opensees as ops
import numpy as np
import pandas as pd
import warnings
#%%

def run_nlth_analysis_on_sdof_ops_py(capacity_curve,gmr,damping,degradation):

      cap_df=pd.DataFrame(capacity_curve)
```

Currently damping is introduced with the Rayleigh command and assumed to be Mass proportional

Damping

master ▾ VMTK-Vulnerability-Modellers-ToolKit / analysis / nlth_on_sdof.py / <> Jump to ▾ Go to file ···

CatalinaYepes start clean up — Latest commit e1518ef on Feb 26, 2021 — History

1 contributor

154 lines (122 sloc) | 5.01 KB — Raw | Blame

```
1   # -*- coding: utf-8 -*-
2   """
3   Created on Fri Jan 17 08:55:51 2020
4
5   @author: LuisMartins
6   """
7
8
9   import openseespy.opensees as ops
10  import numpy as np
11  import pandas as pd
12  import warnings
13  #%%
14
15  def run_nlth_analysis_on_sdof_ops_py(capaci
16
17        cap_df=pd.DataFrame(capacity_curve)
```

```
mx=1
kx=f_vec[1,0]/d_vec[1,0]
omega=np.sqrt(kx/mx)

alphaM=2*damping*omega
ops.rayleigh(alphaM,0,0,0)
```

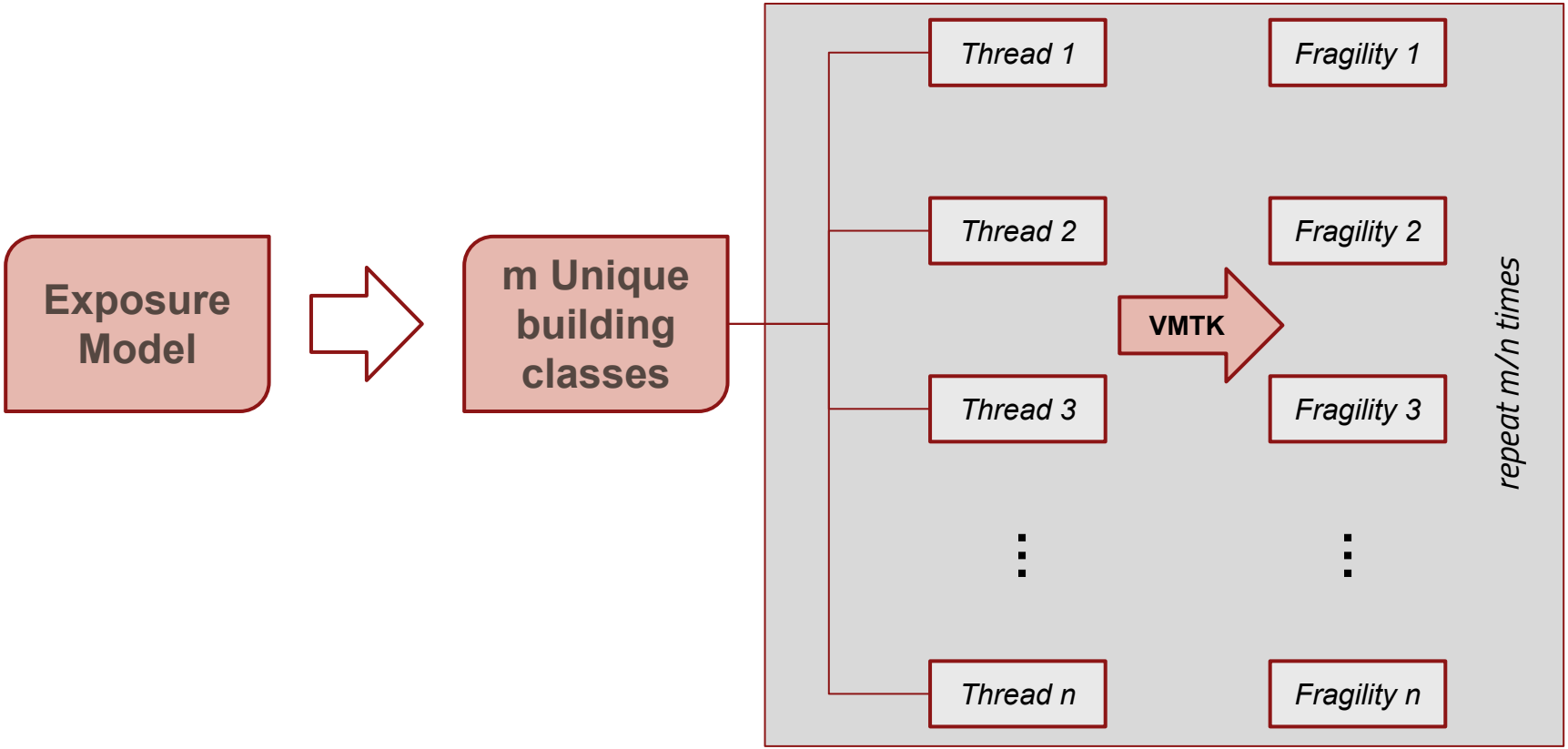
What are other benefits of using the scripts over the GUI?

Ability to parallelize runs for portfolio analysis

One of the most useful capabilities of Python and multi-thread CPU architectures is the ability to parallelize processes effortlessly.

This can be especially useful in portfolio analysis where one wants to assess the performance of multiple building classes.

Ability to parallelize runs for portfolio analysis



Ability to parallelize runs for portfolio analysis

Exposure Model

```python
import multiprocessing

# ---
def function_to_parallelize(args):

    # This function is the portion of the workflow one wants to
    # parallelize

    return True

# ---
if __name__ == '__main__':

    thread_count=multiprocessing.cpu_count()

    pool = multiprocessing.Pool(processes=thread_count)

    pool.map(function_to_parallelize,args)

    pool.close()
```

Fragility 1

Fragility 2

Fragility 3

⋮

Fragility n

repeat m/n times

Where can I find some examples of Python's scripting capabilities applied to the VMTK?

VMTK example script

master ▾ VMTK-Vulnerability-Modellers-ToolKit / demonstration_files / computation_scripts_example.py / <> Jump to ▾ Go to file ···

lmartins88 further cleanup — Latest commit 3666950 4 days ago — History

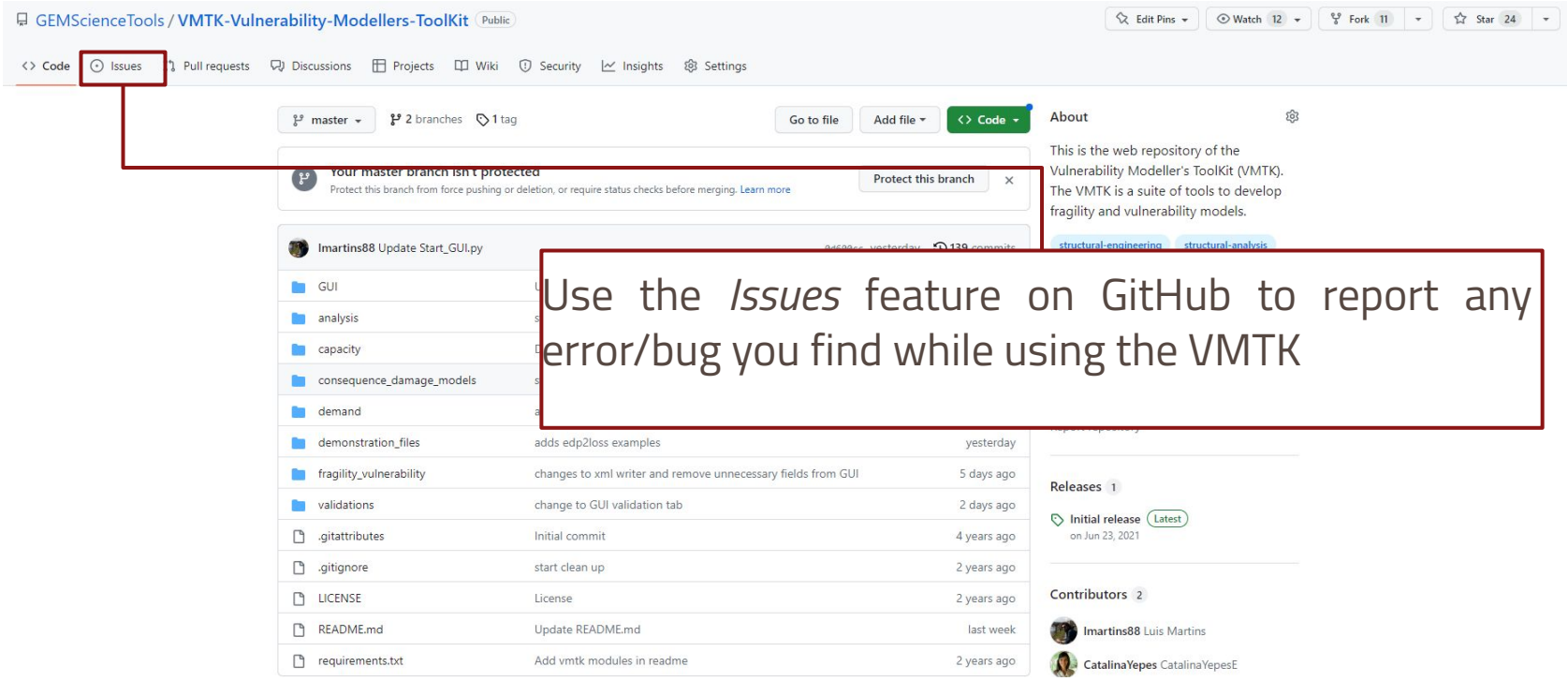
2 contributors

106 lines (82 sloc) | 5.65 KB — Raw | Blame

```
1  # -*- coding: utf-8 -*-
2  """
3  Created on Fri Feb 14 09:34:16 2020
4
5  @author: LuisMartins
6  """
7  #%% import rmtk modules
8  import capacity_start as cap
9  import select_gmrs_start as selgmrs
10 import fragility_vulnerability_start as fvb
11 import nlth_on_sdof as nlth
12
13 #%% import other modules
14 import os
15 import numpy as np
16 import glob
```

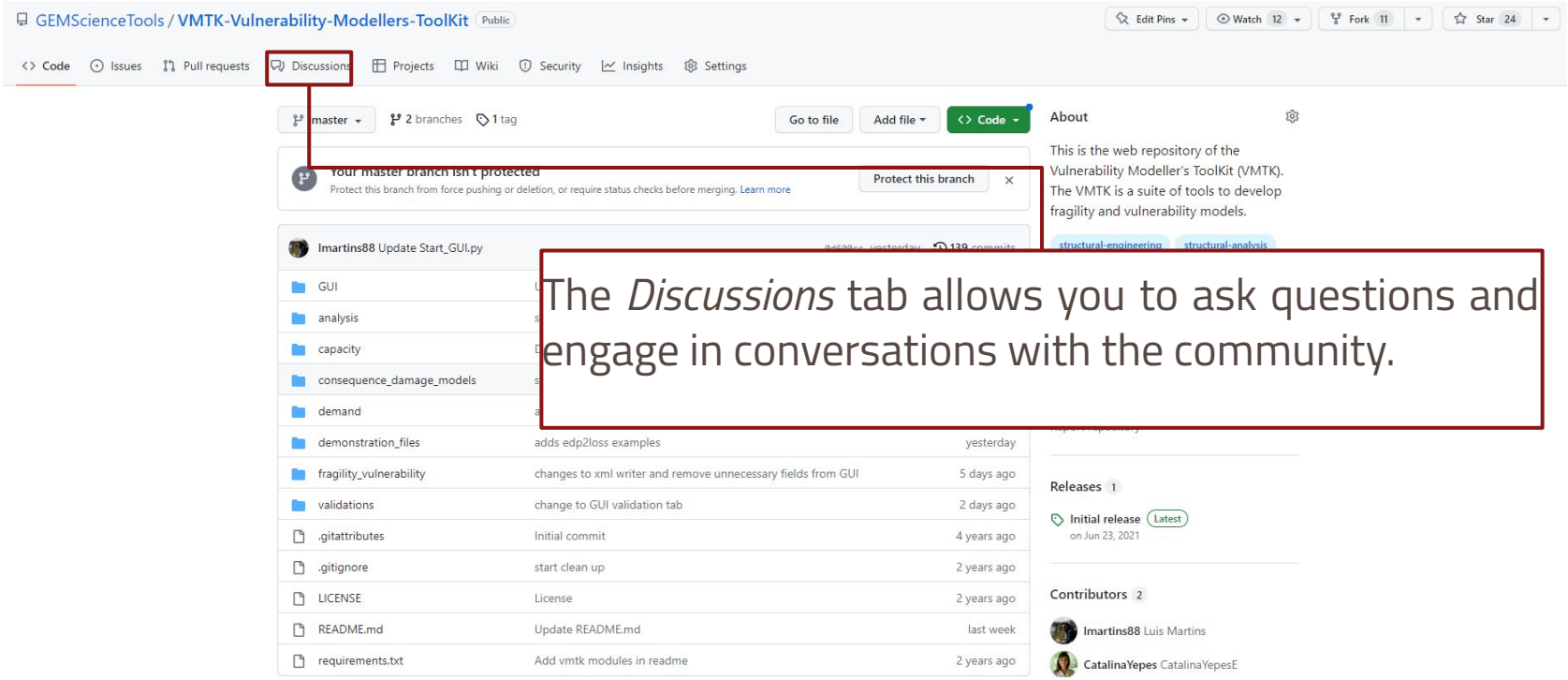
How can I contribute to the VMTK?

Help us improve



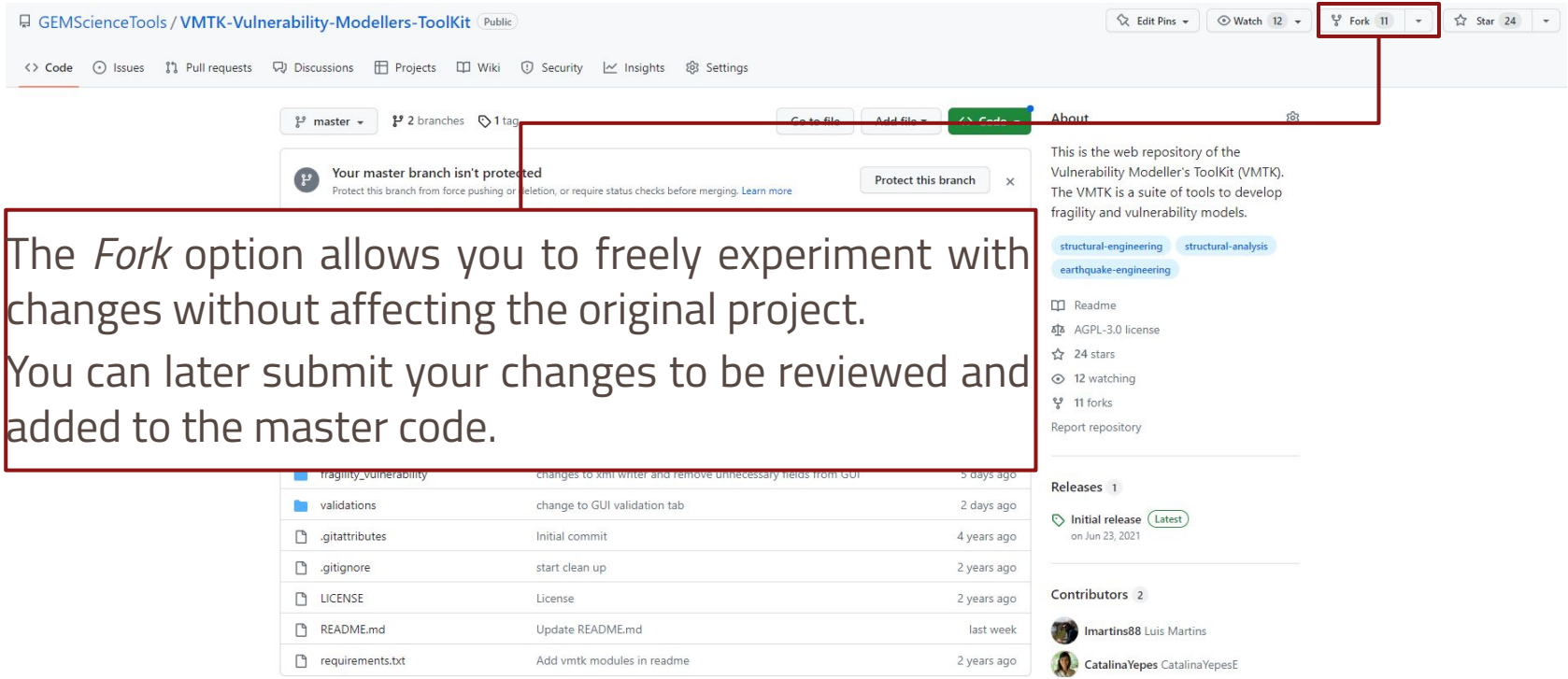
Use the *Issues* feature on GitHub to report any error/bug you find while using the VMTK

Help us improve



The *Discussions* tab allows you to ask questions and engage in conversations with the community.

Help us improve



The *Fork* option allows you to freely experiment with changes without affecting the original project.

You can later submit your changes to be reviewed and added to the master code.

Thank you!

Please attribute to the GEM Foundation with a link to:
https://www.globalquakemodel.org

Except where otherwise noted, this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (CC BY-NC-ND 4.0)
https://creativecommons.org/licenses/by-nc-nd/4.0/