Modellieren und Erdbebenbemessung von einem komplexen Gebäude mit RFEM und Python

Studiengang: Bachelor of Science in Bauingenieurwesen
Fachgebiet: Baumanagement und Digitales Bauen
Betreuer*in: Prof. Dr. Adrian Wildenauer
Experte: Louis Trümpler

Für die Erdbebenbemessung von einem Gebäude wird in der Praxis oft das Antwortspektrenverfahren angewendet. Um die Modellierung und Bemessung eines Gebäudes in RFEM zu vereinfachen wird dafür eine Software mit Python entwickelt.

Ausgangslage

Werden für die Erdbebenbemessung gewisse geometrische Bedingungen nicht eingehalten, muss nach Norm SIA 261 das Antwortspektrenverfahren angewendet werden. Dies wird in der Praxis oft mit einem Statikprogramm durchgeführt. Weil dafür die Modellierung sehr zeitaufwändig ist, wurde ein Programm geschrieben.

Zielsetzung

Das Ziel der vorliegenden Bachelorthesis ist das Entwickeln eines Programms mit der Programmiersprache Python, das die Modellierung und Bemessung des Gebäudes grösstenteils automatisch ausführt. Beim Start des Programms soll eine grafische Benutzeroberfläche geöffnet werden, indem der Benutzer die nötigen Eingaben für die Modellierung und Bemessung eingeben kann. Beim Ausführen des Programms findet ein automatisierter Datenaustausch zwischen Excel und dem Statikprogramm RFEM statt.

Ergebnisse

Das Ergebnis dieser Arbeit ist ein benutzerfreundliches Programm, das schnell und zuverlässig das Gebäude modelliert, der iterative Berechnungsprozess bis zur gewünschten Grundschwingzeit ausführt und die Ergebnisse der Berechnungen in einem Excel ausgibt.

Marco Gerber

```python
for d, (kategorie, name, gf) in enumerate(lastfaelle, 1):

    print(f"Bearbeite Lastfall {d}: {kategorie}")

    if kategorie == 'Eigengewicht + Auflast':
        LoadCase(1, kategorie, [True, 0, 0, 1], ActionCategoryType.ACTION_CATEGORY_PERMANENT_G)
    elif kategorie == 'Nutzlast Kat. A':
        LoadCase(2, kategorie, [False], ActionCategoryType.ACTION_CATEGORY_IMPOSED_LOADS_CATEGORY_A_DOMESTIC_RESIDENTIAL_AREAS_QI_A)
    elif kategorie == 'Nutzlast Kat. B':
        LoadCase(3, kategorie, [False], ActionCategoryType.ACTION_CATEGORY_IMPOSED_LOADS_CATEGORY_B_OFFICE_AREAS_QI_B)
    elif kategorie == 'Nutzlast Kat. C':
        LoadCase(4, kategorie, [False], ActionCategoryType.ACTION_CATEGORY_IMPOSED_LOADS_CATEGORY_C_CONGREGATION_AREAS_QI_C)
    elif kategorie == 'Nutzlast Kat. D':
        LoadCase(5, kategorie, [False], ActionCategoryType.ACTION_CATEGORY_IMPOSED_LOADS_CATEGORY_D_SHOPPING_AREAS_QI_D)
    elif kategorie == 'Nutzlast Kat. E':
        LoadCase(6, kategorie, [False], ActionCategoryType.ACTION_CATEGORY_IMPOSED_LOADS_CATEGORY_E_STORAGE_AREAS_QI_E)
    elif kategorie == 'Nutzlast Kat. H':
        LoadCase(7, kategorie, [False], ActionCategoryType.ACTION_CATEGORY_IMPOSED_LOADS_CATEGORY_H_ROOFS_QI_H)
    elif kategorie == 'Schnee':
        LoadCase(8, kategorie, [False], ActionCategoryType.ACTION_CATEGORY_SNOW_ICE_LOADS_QS)
    else:
        print(f"Unbekannte Lastfall-Kategorie: {kategorie}")
```

Python Code zum Einfügen der Lastfälle in RFEM