

**Dokumentation diverser Programmergänzungen  
zu *Cedrus* (*Cubus AG*) nach SIA 262:2013**

# Querkraftanalyse von FE-Elementen und Übersichtsgrafik für Pfahlausnutzungen

Lucas Maul

Luzern, 2023



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>5</b>
<b>2</b>	<b>Installation</b>	<b>7</b>
2.1	Python-Packages . . . . .	7
2.2	GitHub Download . . . . .	7
2.3	Neuinstallation von Python mittels Anaconda . . . . .	7
2.4	Neuinstallation von Git . . . . .	8
2.5	Erstellen von Berechnungen mit einer IDE . . . . .	8
2.6	Tipps & Tricks . . . . .	8
<b>3</b>	<b>Querkraftnachweise an FE-Elementen</b>	<b>11</b>
3.1	Berechnungsmodell . . . . .	11
3.2	Anwendungsbeispiel . . . . .	13
<b>4</b>	<b>Übersichtsgrafik Pfahlausnutzung</b>	<b>21</b>
4.1	Berechnungsmodell . . . . .	21
4.2	Anwendungsbeispiel . . . . .	21
<b>Literatur</b>		<b>23</b>
	Publikationen . . . . .	23
	Normen . . . . .	23



# 1 Einleitung

Häufig wird in der Bauingenieurpraxis ein Pauschalwert des Querkraftwiderstandes abhängig von der Plattenstärke angenommen und die Stellen, welchen diesen Pauschalwert überschreiten, mittels separater Software detaillierter auf ihren Querkraftwiderstand untersucht. Dabei wird die maximale Kontrolldehnung  $\varepsilon_v$  angesetzt, welche bei 100 % Biegeausnutzung erreicht wird. Dieser Ansatz ist sehr konservativ, weil die effektive Kontrolldehnung in den meisten Fällen durch eine niedrigere Biegeausnutzung kleiner ist. Andererseits kann der Pauschalansatz auch zu optimistisch sein, weil die Einflüsse abweichender Hauptquerkräfte unbeachtet bleiben. Somit kann der Eindruck, der durch einen Pauschalwiderstand entsteht, täuschen. Durch die hier vorliegende Programmierung können die einzelnen Biegeausnutzungen jedes FE-Elementes eines Netzes automatisiert bestimmt und der Querkraftnachweis für jedes FE-Element einer Cedrus-Platte in wenigen Sekunden geführt werden. Das Programm kann die problematischen Elemente farblich markieren und so direkt aufzeigen, wo das Modell definitiv Probleme hat. So kann der Eindruck über den Querkraftwiderstand einer Platte geschärft werden. Dazu wird die tabellarische Ausgabe der Schnittkräfte einer *Resultatkombination* aus Cedrus in Excel kopiert und mittels Python ausgewertet.

Zudem gibt Cedrus bei einer Bodenplatte die Stützenreaktionen der Pfähle grafisch komplett in Blau aus. So ist es schwer, sich direkt einen Eindruck von der Pfahlausnutzung der einzelnen Pfähle zu bilden. Deshalb wurde eine ergänzende farblich abgestufte grafische Ausgabe programmiert, welche optional noch ein Histogramm mitliefert, wie die verschiedenen Ausnutzungen über die Gesamtzahl der Pfähle verteilt sind.

Es wurde versucht, den Prozess möglichst simpel und schnell zu halten. Der Datentransfer von Cedrus via Excel in Python ist ohne Formatierungsaufwand.



## 2 Installation

### 2.1 Python-Packages

Alle Installationen und benötigten Programme zur Anwendung der Python-Codes sind vollkommen gratis im Internet über offizielle Wege erhältlich.

Die Python-Programmierung bedient sich einiger bereits programmierter Python-Bibliotheken, welche im Folgenden aufgelistet sind:

```
## Standard Packages (bei Anaconda enthalten)
- pandas # Tabellenkalkulationen
- numpy # Lineare Algebra
- matplotlib # Grafische Ausgabe
```

Am besten installiert man Anaconda, bei dem alle benötigten Pakete, sowie die Python-Umgebung automatisch mitgeliefert werden.

Sind die Pakete installiert, können die Dateien von *Git-Hub* heruntergeladen werden. Dazu muss eine Git Installation vorhanden sein. Für Git kann der in [Abschnitt 2.2](#) aufgeführte Befehl in der Windows Eingabeaufforderung (Win + R, 'cmd', Enter) oder im Mac Terminal (Shift + Space, 'term', Enter) in dem Verzeichnis ausgeführt werden, in das die Python-Dateien heruntergeladen werden sollen.

Sollte der Anwender keine Ahnung von den oben genannten Angaben haben, und noch nie in Python programmiert haben, wird es empfohlen ab Schritt [2.3](#) die Programme nacheinander zu installieren und auszuführen.

### 2.2 GitHub Download

```
git clone https://github.com/LucasMaul/05-Pfahlcoring-Cedrus-Cubus.git
```

```
git clone https://github.com/LucasMaul/06-Elementnachweise-Cubus-CEDRUS.git
```

### 2.3 Neuinstallation von Python mittels Anaconda

Anaconda kann unter folgender URL heruntergeladen und installiert werden:

<https://www.anaconda.com/download>

### 2.4 Neuinstallation von Git

Git kann unter folgender URL heruntergeladen werden:

<https://git-scm.com/download/win>

Nach der Installation von Git kann der Befehl in [Abschnitt 2.2](#) zum Download in der Windows-Eingabeaufforderung oder im Mac-Terminal eingegeben werden, um an die Python Dateien der in dieser Arbeit beschriebenen Programmierung zu kommen.

### 2.5 Erstellen von Berechnungen mit einer IDE

Die Python Programmierung ist auf mehrere Python-Dateien aufgeteilt. Die Berechnungsanweisung und Steuerung des gesamten Codes findet im *main* File statt. Es sind in der Git-Repository standardmässig mehrere Berechnungsbeispiele, die als *main*-Dateien benannt sind, aufgeführt.

Es wird empfohlen zur einfachen Bearbeitung dieser *main*-Files eine *IDE* (Editor) herunterzuladen und zu installieren. Als sehr handlich und umfangreich hat sich *Visual Studio Code* (kurz *VS Code*) etabliert, das mit tausenden PlugIns (Erweiterungen) von einer grossen Community gespeist wird.

*VS Code* kann unter folgender URL heruntergeladen werden:

<https://code.visualstudio.com/download>

Nach der Installation und einem Neustart von Anaconda, wird *VS Code* idealerweise über Anaconda geöffnet. Wenn der Python Code das erste Mal ausgeführt wird, muss die Umgebung gewählt werden. Bei dieser sollte *\* base conda* selektiert werden.

### 2.6 Tipps & Tricks

#### 2.6.1 Python lernen

Ein vollständiger empfehlenswerter gratis Einstiegskurs in *Python* in Deutsch findet sich unter:

<https://youtube.com/playlist?list=PLNmsVeXQZj7q0ao69AIogD94oBgp3E9Zs>



### 2.6.2 Plugins für *VS Code* in *Python* und $\text{\LaTeX}$

Zur Arbeit mit Python mit *VS Code* werden folgende PlugIns empfohlen:

- Pep/8 Assembly Highlighter (Fehlerprüfung der Python-Programmierung, sogenanntes 'linting')
- Pylance (gibt den Dateien schöne Icons)
- Pylint (linting)
- Python (liniting)

Für die Nutzung von  $\text{\LaTeX}$  werden folgende PlugIns bei *VS Code* empfohlen:

- LaTeX Workshop
- LaTeX language support
- LaTeX Utilities
- German Language Pack for Visual Studio Code (Rechtschreibprüfung beim Schreiben in Deutsch)

### 2.6.3 Darstellung von Programmiercodes in $\text{\LaTeX}$

Zur automatischen Darstellung der Python Programmierung wurde eine *minted* Umgebung verwendet. Nähere Informationen dazu unter:

[https://www.overleaf.com/learn/latex/Code\\_Highlighting\\_with\\_minted](https://www.overleaf.com/learn/latex/Code_Highlighting_with_minted)



## 3 Querkraftnachweise an FE-Elementen

### 3.1 Berechnungsmodell

Die ausgeführten Berechnungen stützen sich im Wesentlichen auf der Norm SIA 262:2013 Ziffer 4.3.3.2 [2] und dem Vorlesungsskript *Bemessung von Stahlbetonplatten mit FEM* [1] der Hochschule Luzern.

Der normative Nachweis der Plattenquerkraft wird wie folgt geführt:

$$v_{Rd} = k_d \cdot \tau_{cd} \cdot d_v \geq v_d \quad \text{mit} \quad d_v \leq \frac{d_{sx} + d_{sy}}{2} \quad (3.1)$$

Dabei beschreiben  $d_{sx}$  und  $d_{sy}$  die jeweiligen statischen Höhen der Bewehrungslagen in x und y Richtung. Die Querkräfte werden jeweils mit den dazugehörigen Biegemomenten gemäss Ziffer 4.3.3.2.4 in ihrer Hauptrichtung untersucht. Somit wird Gl. 3.1 in Gl. 3.2 übergeführt.

$$v_{Rd}(\varphi_{0d}) = k_d(\varphi_{0d}) \cdot \tau_{cd} \cdot d_v \geq v_{0d} \quad (3.2)$$

mit

$$v_{0d} = \sqrt{v_{xd}^2 + v_{yd}^2} \quad (3.3)$$

$$\varphi_{0d} = \arctan\left(\frac{v_{yd}}{v_{xd}}\right) \quad (3.4)$$

$$k_d = \frac{1}{1 + e_v \cdot d_v \cdot k_g \cdot \zeta} \quad (3.5)$$

$$k_g = \frac{48}{16 + D_{max}} \quad (3.6)$$

$$\zeta = \frac{1}{\sin^4(\varphi_{0d}) + \cos^4(\varphi_{0d})} \quad (3.7)$$

$$e_v = \frac{f_{sd}}{E_s} \cdot \frac{m_{nd}(\varphi_{0d})}{m_{nRd}(\varphi_{0d})} \quad (3.8)$$

### 3 Querkraftnachweise an FE-Elementen

Die Biegemomente  $m_{nd}(\varphi_{0d})$  ergeben sich aus den in *Cedrus* ermittelten Schnittgrößen:

$$m_{nd}(\varphi_{0d}) = m_{xd} \cdot \cos^2(\varphi_{0d}) + m_{yd} \cdot \sin^2(\varphi_{0d}) + 2 \cdot m_{xyd} \cdot \sin(\varphi_{0d}) \cdot \cos(\varphi_{0d}) \quad (3.9)$$

$$m_{nRd}(\varphi_{0d}) = m_{xRd} \cdot \cos^2(\varphi_{0d}) + m_{yRd} \cdot \sin^2(\varphi_{0d}) \quad (3.10)$$

wobei sich die Biegewiderstände  $m_{nRd}(\varphi_{0d})$  aus den Biegewiderständen  $m_{xRd}$  und  $m_{yRd}$  aus der Querschnittsanalyse für einen 1000 mm breiten rechteckigen Stahlbetonquerschnitt zusammenstellen:

$$m_{xRd} = a_s \cdot f_{sd} \cdot \left( d_{sx} - \frac{a_s \cdot f_{sd}}{2 \cdot 1000 \cdot f_{cd}} \right) \quad (3.11)$$

$$m_{yRd} = a_s \cdot f_{sd} \cdot \left( d_{sy} - \frac{a_s \cdot f_{sd}}{2 \cdot 1000 \cdot f_{cd}} \right) \quad (3.12)$$

## 3.2 Anwendungsbeispiel

### 3.2.1 Programmdateien herunterladen

Wenn *Git* gemäss [Abschnitt 2.4](#) installiert ist, geht man am besten wie folgt vor:

1. Eingabeaufforderung/Terminal öffnen:

mac:  
command + leertaste > "terminal" eingeben > enter

windows:  
win + r > "cmd" eingeben > enter

2. In der Eingabeaufforderungen/Terminal auf z.B. auf den Desktop navigieren:

mac & windows:  
cd desktop

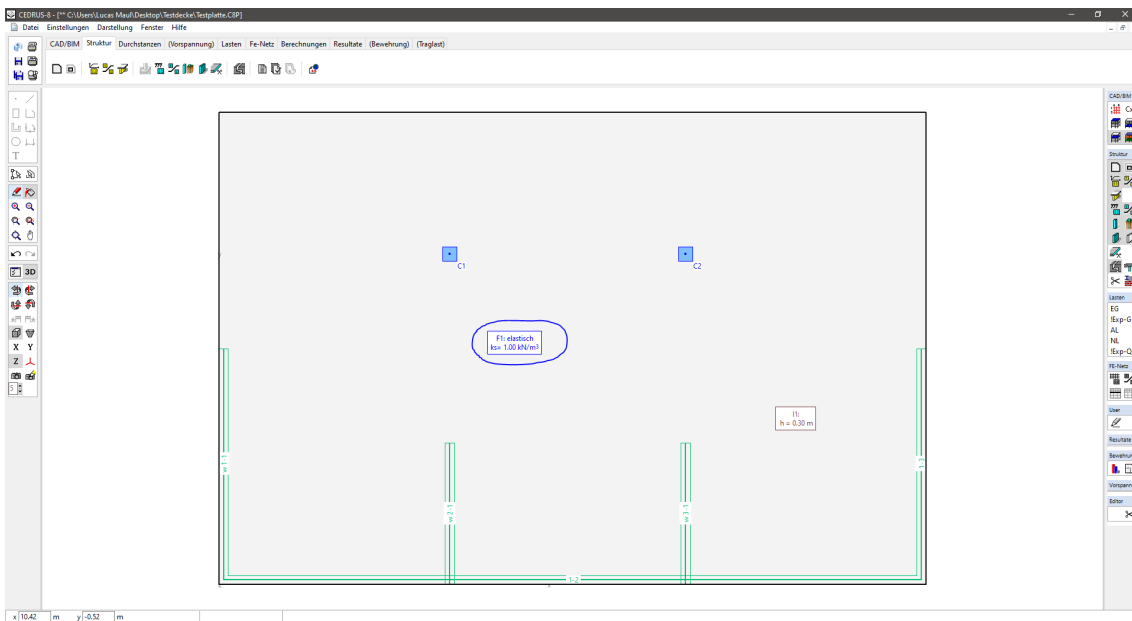
3. In der Eingabeaufforderung folgenden Befehl ausführen:

mac & windows:  
git clone https://github.com/LucasMaul/06-Elementnachweise-Cubus-CEDRUS.git

Auf dem Desktop sollte nun ein Ordner mit den Namen *06-Elementnachweise-Cubus-CEDRUS* erstellt worden sein.

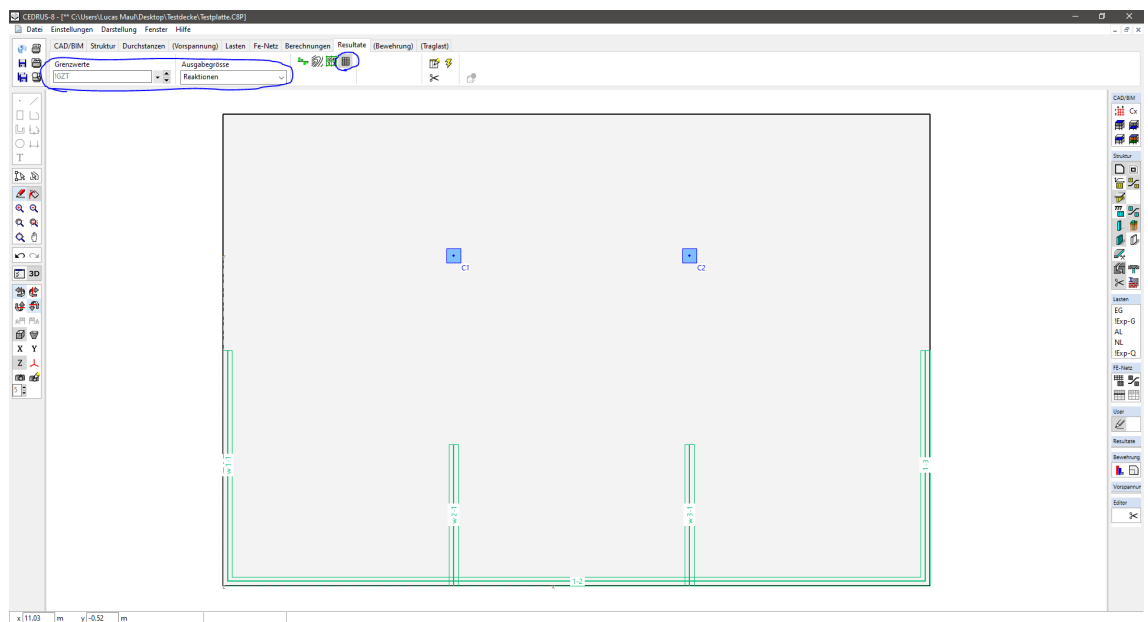
### 3.2.2 Cedrus Daten extrahieren

Für die grafische Ausgabe der Querkraftnachweise werden die Koordinaten der einzelnen FE-Elemente benötigt. *Cedrus* selbst scheint keine direkt eingebaute Möglichkeit zu haben, diese direkt auszulesen. Deshalb wird ein Trick angewendet. Im Strukturregister wird temporär ein globales Flächenlager eingeführt.

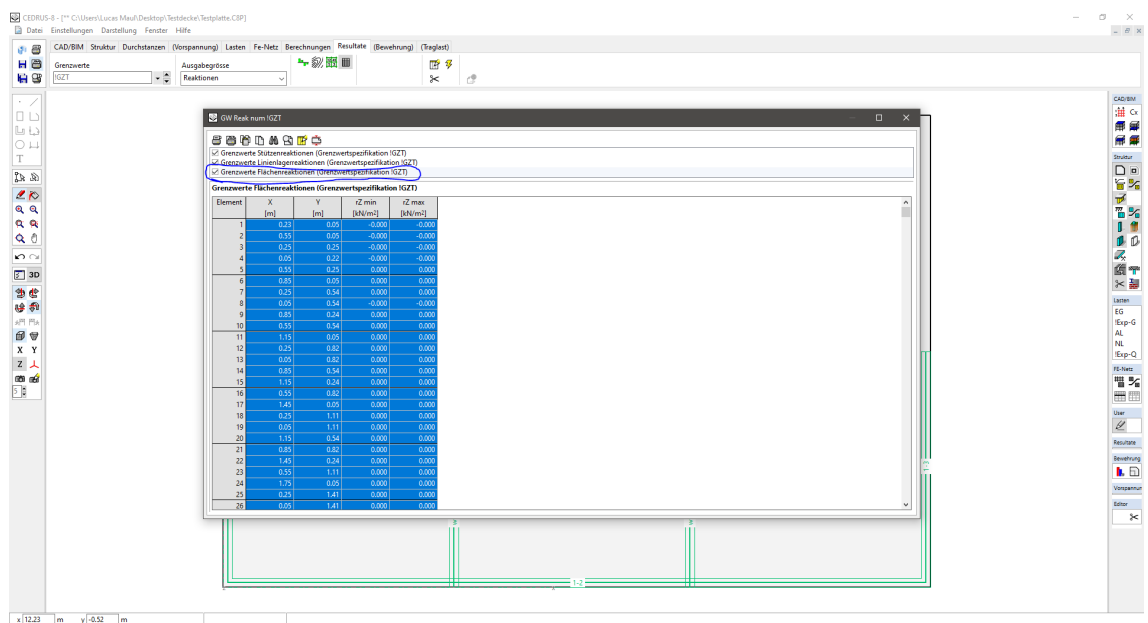


### 3 Querkraftnachweise an FE-Elementen

Anschliessend werden im Resultateregister im Grenzwert GZT! (spielt keine Rolle welcher) die Reaktionen tabellarisch ausgelesen.



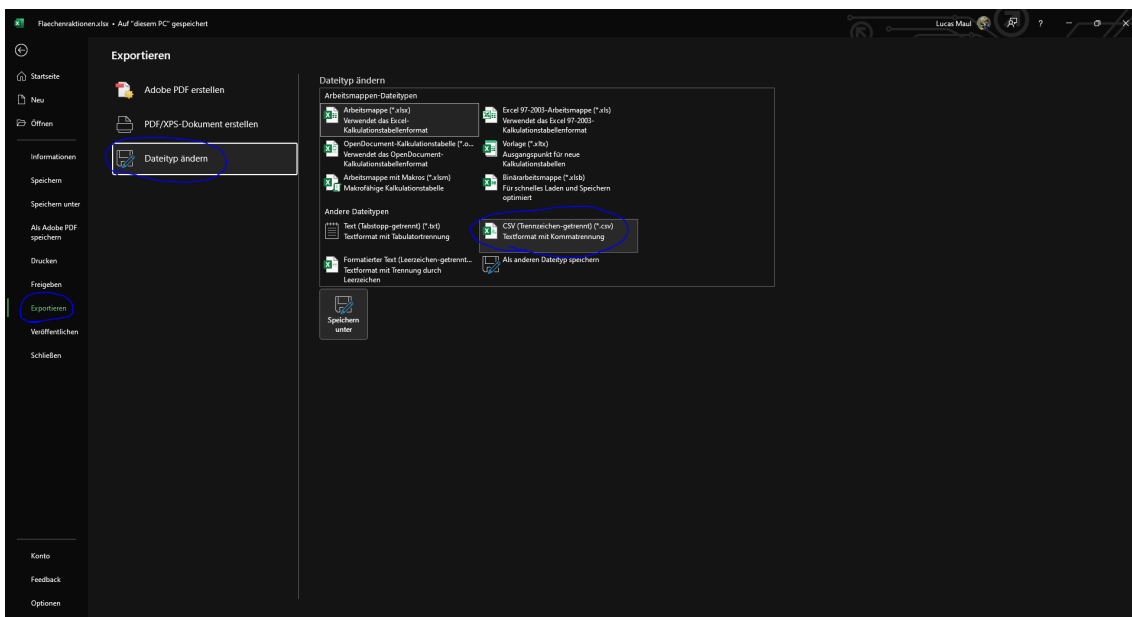
Die Tabelle *Grenzwerte Flächenreaktionen (Grenzwertspezifikation !GZT)* kann mit *Strg + a* ausgewählt und mit *Strg + c* in die Zwischenablage kopiert werden.



Nun können die Tabellenwerte in eine Exceldatei eingefügt werden (*Strg + v*), um diese in das CSV-Dateiformat zu konvertieren. In den heruntergeladenen Dateien befindet sich bereits eine Exceldatei namens *Flaechenreaktionen.xlsx*, die dafür gedacht ist. Es ist keine Überschriftenzeile notwendig.

	A	B	C	D	E	F	G	H	I	J
1	14.8	9.85	0.005	0.007						
2	12.89	8.85	0.005	0.011						
3	12.99	9.55	0.005	0.011						
4	13.28	9.25	0.005	0.011						
5	13.58	8.95	0.005	0.01						
6	13.87	8.65	0.005	0.01						
7	14.16	8.35	0.004	0.009						
8	14.46	8.05	0.004	0.008						
9	14.8	7.75	0.004	0.008						
10	12.99	9.85	0.006	0.012						
11	13.28	9.55	0.006	0.011						
12	13.58	9.25	0.005	0.011						
13	13.87	8.95	0.005	0.01						
14	14.16	8.65	0.005	0.01						
15	14.46	8.35	0.004	0.009						
16	14.8	8.05	0.004	0.009						
17	13.28	9.85	0.006	0.012						
18	13.58	9.55	0.006	0.012						
19	13.87	9.25	0.005	0.011						
20	14.16	8.95	0.005	0.011						
21	14.46	8.65	0.005	0.01						
22	14.8	8.35	0.005	0.01						
23	13.58	9.85	0.006	0.013						
24	13.87	9.55	0.006	0.012						
25	14.16	9.25	0.006	0.012						
26	14.46	8.95	0.005	0.011						
27	14.8	8.65	0.005	0.011						
28	13.87	9.85	0.006	0.013						
29	14.16	9.55	0.006	0.013						
30	14.46	9.25	0.006	0.012						
31	14.8	8.95	0.006	0.011						
32	14.16	9.85	0.006	0.013						
33	14.46	9.55	0.006	0.013						
34	14.8	9.25	0.006	0.012						
35	14.46	9.85	0.007	0.014						
36	14.8	9.55	0.005	0.013						
37	14.8	9.85	0.007	0.014						

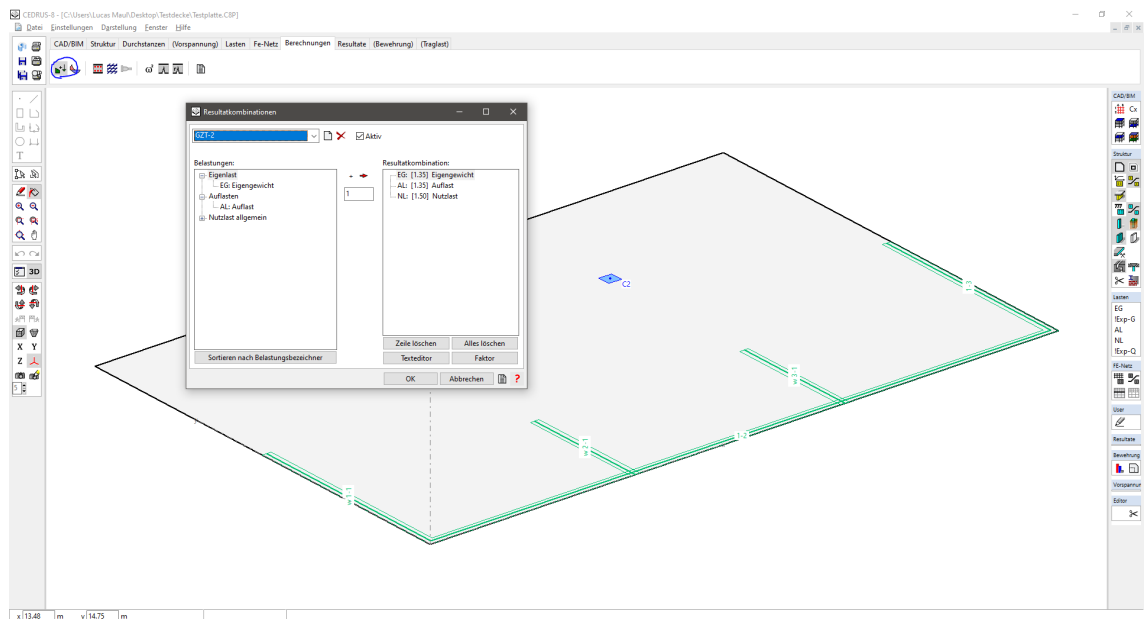
Danach die Tabelle als CSV Tabelle exportieren und idealerweise direkt in den Ordner speichern, in dem die Dateien heruntergeladen wurden (bzw. sich das *xlsx*-File bereits befindet).



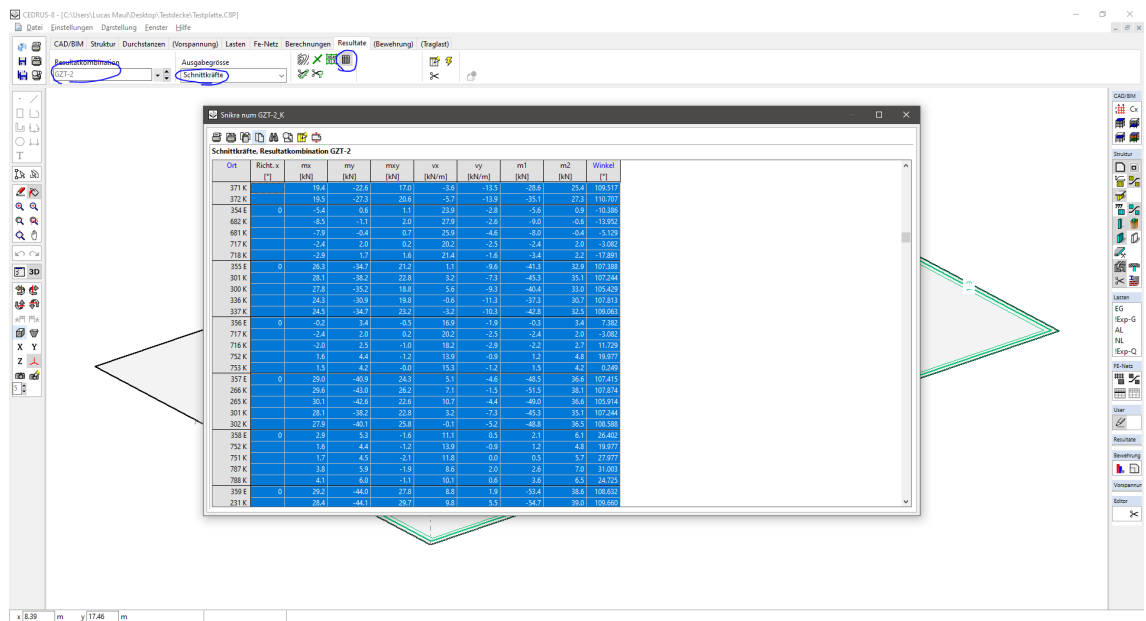
Nun zurück zu *Cedrus*. Dort muss das Flächenlager wieder gelöscht werden. Das FE-Netz sollte sich i.d.R. exakt gleich bilden, wie es zuvor ohne Flächenlager war. Um an die einzelnen Schnittkraftkomponenten zu kommen, muss händisch eine Resultatkombination erstellt werden. Der automatisch erstellte Grenzwerte der Tragsicherheit *!GZT* kumuliert verschiedene Lastfälle und ermittelt die *Umhüllende*. Für den Querkraftnachweis sind allerdings die Schnittgrößen einer Gleichgewichtslösung notwendig, andernfalls vergleicht man Äpfel mit Birnen (Querkraftbeanspruchung mit falsch zugeordneter Biegung). Ist nicht pauschal beurteilbar, welcher Lastfall an welcher Stelle zu maximaler Querkraft-

### 3 Querkraftnachweise an FE-Elementen

bzw. Biegebeanspruchung führt, kann man mehrere Resultatkombinationen erstellen, diese jeweils in eigene *csv*-Tabellen speichern und einzeln in *Python* auswerten. Die händische Datenextraktion aus *Cedrus* kann allerdings je nach Anzahl Lastkombinationen sehr aufwendig werden.



Nun werden die Schnittkraftkomponenten in *Cedrus* im Resultatregister als Tabelle ausgegeben und anschliessend wieder in Excel kopiert und als *csv* gespeichert.



Jetzt sind die für Python benötigten Daten vorhanden und es kann in der *IDE* (in diesen Beispiel in *Visual Studio Code* - kurz *vscode*) weitergearbeitet werden. Es sollten sich nun folgende Dateien im besagten Ordner, in dem gearbeitet werden soll, befinden:



Elementnachweise.ipynb	15.09.2023 19:30	Jupyter-Quelldatei	93 KB
Elementnachweise.py	15.09.2023 19:30	Python-Quelldatei	13 KB
Flaechenreaktionen.xlsx	16.09.2023 18:04	Microsoft Excel-A...	44 KB
Flaechenreaktionen.csv	16.09.2023 18:10	Microsoft Excel-C...	33 KB
README.md	15.09.2023 19:30	Markdown-Quelld...	1 KB
Schnittkraefte.csv	15.09.2023 19:30	Microsoft Excel-C...	4'952 KB
Schnittkraefte.xlsx	15.09.2023 19:30	Microsoft Excel-A...	5'857 KB

### 3.2.3 Daten in Python auswerten

Ist *Visual Studio Code* geöffnet, kann der besagte Ordner mit den Dateien geöffnet werden, indem man in *vscode* oben im Register auf "Datei» Ordner öffnen"(alternativ unter Windows *Strg + k + o*) geht. Anschliessend ist es vermutlich am praktischsten, in einer Jupyter-Datei zu arbeiten. In dieser werden die Ergebnisse direkt unter dem Befehl ausgegeben. Zudem sind die Python-Befehle sehr übersichtlich einsehbar. Die Jupyter-Datei ist standardmässig im besagten Ordner schon erstellt. Falls man selbst eine neue anlegen möchte, kann man oben im Register in *vscode* auf "Datei» "neue Datei"(alternativ *Strg + alt + windows + n*) gehen und dieser die Dateiendung *.ipynb* geben.

```

# Imports the "Platte" object from the "Elementnachweise.py" file into this notebook
from Elementnachweise import Platte

# Saves an "Platte" object with specific attributes into the variable "ceiling_1". "Richtung" describes the destination of the 4th reinforcement layer.
ceiling_1 = Platte(filename='Schnittkraefte.csv', filename_reaction='Flaechenreaktionen.csv', c_nom_mm=30, d_nom=300, Richtung='x')
d4_mm=12, a4_mm=150, d3_mm=12, a3_mm=150, s_mm=0, f_cd_Nmm2=16.5, tau_cd_Nmm2=1.00, Dmax_mm=32, f_sd_Nmm2=435, f_s_Nmm2=205000)
# Multiple variables with different attributes can be created
ceiling_2 = Platte(filename='Schnittkraefte.csv', filename_reaction='Flaechenreaktionen.csv', c_nom_mm=30, d_nom=300, Richtung='x',
d4_mm=12, a4_mm=150, d3_mm=12, a3_mm=150, s_mm=0, f_cd_Nmm2=16.5, tau_cd_Nmm2=1.00, Dmax_mm=32, f_sd_Nmm2=435, f_s_Nmm2=205000)

# only last command row is automatically printed.
ceiling_1.a_3_Rd() # getting the bending resistance of the third layer
ceiling_1.a_4_Rd() # getting the bending resistance of the fourth layer
ceiling_1.a_x_Rd() # getting the bending resistance in x-direction
ceiling_1.a_y_Rd() # getting the bending resistance in y-direction

# shows some info
ceiling_1.print_info()

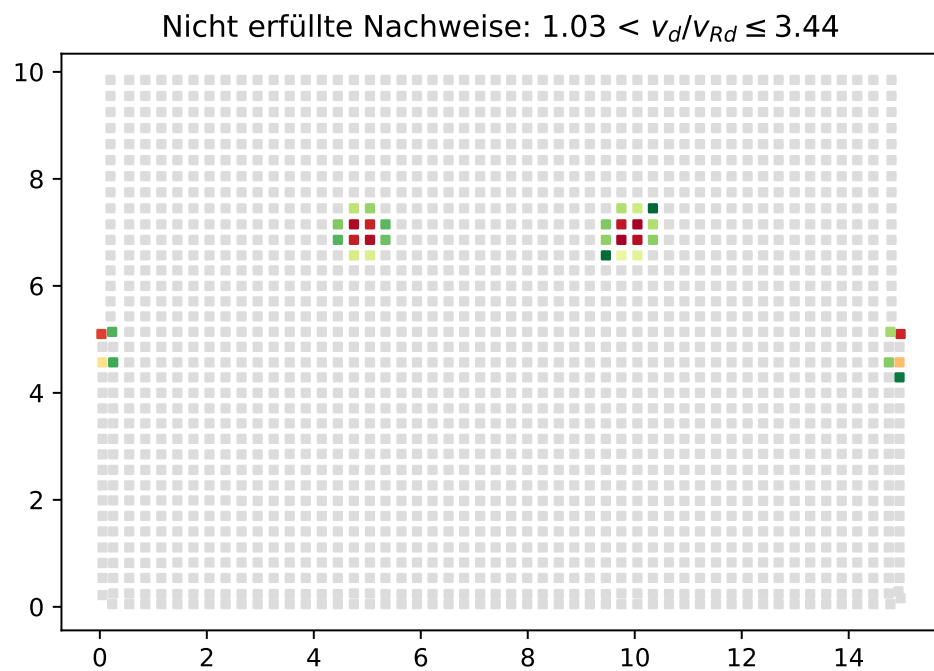
Decke d = 300 mm, c_nom = 30 mm, Einsenkung s = 0 mm
Bewehrung: x 12/150 mm, y 12/150 mm, f_cd = 16.5 N/mm2, f_s = 205000 N/mm2
Beton: f_cd = 16.5 N/mm2, tau_cd = 1.00 N/mm2, D_max = 32 mm

```

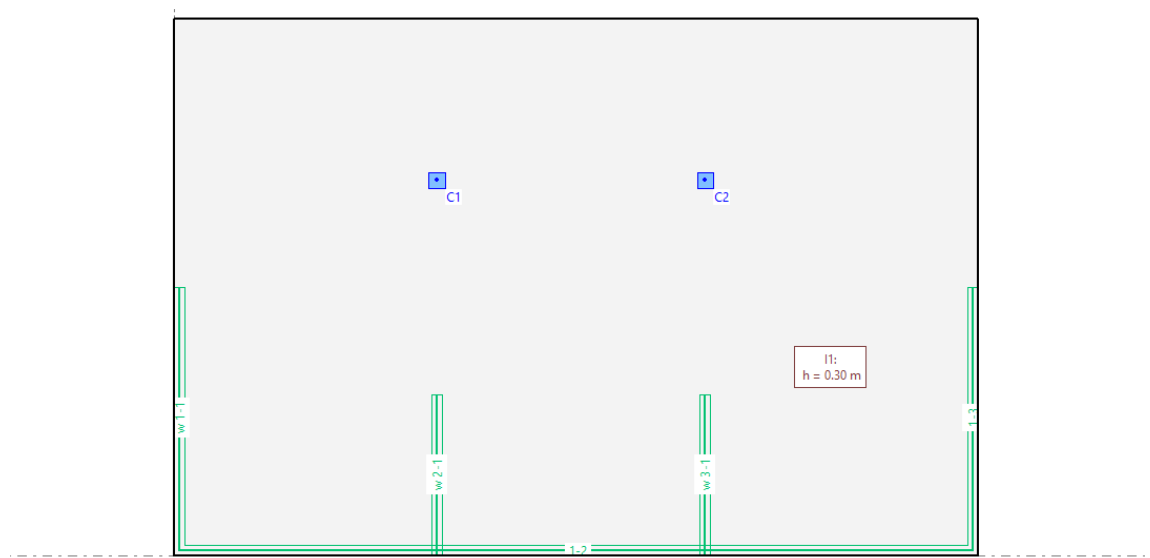
In der Datei *Elementnachweise.ipynb* sind nun einige Analyseschritte erklärt. Für die erste Ausführung des Notebooks ist ein *Kernel* (so etwas wie eine Umgebung) notwendig, in der Programmierung ausgeführt wird. Nun liegt es beim Anwender, mit dem Programm "herumzuspielen". Um eine Zeile in Jupyter auszuführen, drückt man am besten *shift + enter*. Neue Zeilen unterhalb erstellt man mit *b*, oberhalb mit *a*. Zur Wandlung einer Zeile in eine Markdown (Kommentarzeile) kann man *m* verwenden, in eine Codezeile *y*. Weitere Tastenbefehle finden sich hier: <https://code.visualstudio.com/docs/datascience/jupyter-notebooks>.

### 3 Querkraftnachweise an FE-Elementen

Nachfolgend wird eine Platte mittels Python ausgewertet. In Bild 3.1 ist farbig ersichtlich, mit welchen FE-Elementen der Querkraftnachweis nicht gelingt. Die beiden Wandenden im Plattenfeld scheinen bzgl. Querkraft unproblematisch zu sein.



**Bild 3.1:** Auswertung der Querkraftwiderstände in Python. Farbig: gescheiterte Querkraftnachweise



**Bild 3.2:** Ausgewertete Struktur in Cedrus

## 3.2.4 Ausgaben und Attribute

Hier sind noch einige Programmiercodes erklärt:

```
# Verknüpft das Objekt "Platte" der Datei Elementnachweise.py
# mit dem Jupyter-Notebook
from Elementnachweise import Platte

### Attribute ###

Platte = Platte(
    filename_innerforce='Schnittkraefte.csv',
    filename_raction='Flaechenreaktionen.csv', c_nom_mm=30, d_mm=300,
    Richtung='x', d4_mm=12, a4_mm=150, d3_mm=12, a3_mm=150, s_mm=0,
    f_cd_Nmm2=16.5, tau_cd_Nmm2=1.00, Dmax_mm=32, f_sd_Nmm2=435,
    E_s_Nmm2=205000
)

# Dateiname (falls in einem anderen Verzeichnis, vollständiger Pfad + Dateiname)
# der csv-Datei, in der die Schnittkraftwerte abgelegt sind.
filename_innerforce
# Dateiname der csv-Datei, in der die Flächenreaktionen abgelegt sind.
filename_raction
# Bewehrungsüberdeckung nach SIA 262:213
c_nom_mm
# Plattenstärke
d_mm
# Richtung der 4. Bewehrungslage
Richtung
# Durchmesser d und Teilung a der 4. resp. 3. Bewehrungslage
d4_mm
a4_mm
d3_mm
a3_mm
# Einsenkung (z.B. Kopfplatte einer Stütze oder Leitungen in der Decke)
s_mm
# Betondruckfestigkeit nach SIA 262:2013
f_cd_Nmm2
# Betonschubfestigkeit nach SIA 262:2013
tau_cd_Nmm2
# Beton Grösstkorndurchmesser nach SIA 262:2013
Dmax_mm
# Bewehrungsstahlzugfestigkeit nach SIA 262:2013
f_sd_Nmm2
# Elastizitätsmodul des Baustahls nach SIA 262:2013
E_s_Nmm2

### Methoden ###

# Enthält den Biegewiderstand der Platte in Richtung der vierten resp. dritten Lage
Platte.m_3_Rd()
Platte.m_4_Rd()

# Enthält den Biegewiderstand der Platte in x bzw. y Richtung
Platte.m_x_Rd()
Platte.m_y_Rd()

# Gibt die Platteninformationen aus (gut zur Eingabekontrolle geeignet)
```

### 3 Querkraftnachweise an FE-Elementen

```
Platte.print_info()
```

```
# Druckt eine Grafik der FE-Elemente, deren Nachweis gescheitert ist  
Platte.print_Nachweise(Typ='V', save_file=False, markersize=10, bad_only=False,  
    annotation=False, annotation_size=5)  
# Typ entweder V für Querkraft- oder M für Biegeanalyse  
# save_file=True speichert eine png und pdf Datei ins geöffnete Verzeichnis  
# markersize bestimmt die Darstellungsgrösse der einzelnen FE-Elemente  
# annotation=True druckt die Zahlenwerte der misslungenen Nachweise  
# annotation_size bestimmt die grösse der Zahlenwerte  
  
# Gibt die einzelnen Nachweise der FE-Elemente als Tabelle aus  
Platte.Querkraftnachweise_erfuellt(sort_values=True, save_csv=False)  
Platte.Querkraftnachweise_nicht_erfuellt(sort_values=True, save_csv=False)  
Platte.Biegenachweise_erfuellt(sort_values=True, save_csv=False)  
Platte.Biegenachweise_nicht_erfuellt(sort_values=True, save_csv=False)  
# save_csv=True speichert eine csv-Tabelle ins geöffnete Verzeichnis  
# sort_values=False sortiert die Zeichen nach Reihenfolge der FE-Elemente
```

## 4 Übersichtsgrafik Pfahlausnutzung

### 4.1 Berechnungsmodell

### 4.2 Anwendungsbeispiel



# Literatur

## Publikationen

- [1] Spathelf Thoma. *Bemessung von Stahlbetonplatten mit FEM*. Skript zur Vorlesung. 2022.

## Normen

- [2] Schweizerischer Ingenieur- und Architektenverein SIA. *Norm SIA 262:2013 Betonbau*. Zürich. 2013.