

Sprawozdanie z projektu

**Wyznaczenie naprężeń normalnych dla
rozciągania i zginania belki dwuwymiarowej**

Obliczenia Inżynierskie w chmurze

prowadzący: dr inż. Mateusz Żbikowski

Warszawa, 20 stycznia 2025

1. Wprowadzenie i opis projektu

Celem projektu było wyznaczenie naprężeń normalnych dla rozciągania i zginania belki dwuwymiarowej o przekroju prostokątnym utwierdzonej na jednym z końców. Program umożliwia użytkownikowi wprowadzenie danych geometrycznych przekroju, wartości siły normalnej, siły poprzecznej oraz momentu gnącego na swobodnym jej końcu.

W celu przeprowadzenia analiz stworzony został program w języku *Python*. Wykorzystana została maszyna wirtualna w serwisie *Microsoft Azure* przy użyciu programu *FileZilla*, który umożliwił skopiowanie plików, oraz aplikacji *Docker*, który stworzył obraz maszyny.

2. Działanie programu

Na potrzeby projektu stworzony został program w języku *Python*. Analizowanym przypadkiem była pozioma belka dwuwymiarowa o przekroju prostokątnym utwierdzona na jednym z końców, natomiast drugi koniec był swobodny. Na początku użytkownik wprowadza dane geometryczne analizowanej belki, czyli określa szerokość i wysokość przekroju prostokątnego, a następnie długość belki. Kolejnym etapem jest zdefiniowanie wartości siły normalnej, siły poprzecznej i momentu, którymi obciążony został swobodny koniec belki. Wartości dodatnie są zgodne z przyjętą konwencją znaków w równaniach wytrzymałościowych, wartości minusowe mają natomiast przeciwny zwrot.

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  def calculate_stresses():
5      print("\nObliczanie naprężeń normalnych od rozciągania i zginania belki prostokątnej.")
6      #Pozioma belka o prostokątnym przekroju utwierdzona jest z jednej strony,
7      #natomiast drugi jej koniec jest swobodny. Na swobodnym końcu wprowadzana zostaje
8      #siła normalna, siła poprzeczna i moment gnący z konwencją zgodną z wytrzymałością konstrukcji.
9
10     # Dane wejściowe
11     try:
12         b = float(input("Podaj szerokość przekroju prostokątnego (b) [m]: "))
13         h = float(input("Podaj wysokość przekroju prostokątnego (h) [m]: "))
14         if b <= 0 or h <= 0:
15             raise ValueError("Wymiary przekroju muszą być dodatnie!")
16         L = float(input("Podaj długość belki (L) [m]: "))
17         if L <= 0:
18             raise ValueError("Długość belki musi być dodatnia!")
19
20         N = float(input("Podaj wartość siły normalnej na swobodnym końcu belki (N) [N]: "))
21         T = float(input("Podaj wartość siły poprzecznej na swobodnym końcu belki (T) [N]: "))
22         M = float(input("Podaj wartość momentu gnącego na swobodnym końcu belki (M) [Nm]: "))
23
```

Rysunek 1 – Kod programu, który umożliwia wprowadzenie danych przez użytkownika

Kolejnym etapem jest wyznaczenie maksymalnego momentu gnącego, który występuje w utwierdzonym końcu belki za pomocą prostej zależności. Wiedząc, że największe naprężenia normalne występują w tym miejscu, obliczone zostały maksymalne wartości naprężeń od siły normalnej oraz momentu gnącego. Następnie wyznaczone zostały naprężenia całkowite w najbardziej wytężonym miejscu.

```

24 # Wyznaczenie momentu gnącego w utwierdzonym końcu belki
25 M_max = M + T * L
26 print(f"\nMoment gnący w utwierdzeniu belki (M_max): {M_max:.6f} Nm")
27
28 # Pole przekroju
29 A = b * h
30 print(f"Pole przekroju (A): {A:.6f} m^2")
31
32 # Moment bezwładności przekroju względem osi zginania
33 I = (b * h**3) / 12
34 print(f"Moment bezwładności przekroju (I): {I:.6f} m^4")
35
36 # Naprężenie normalne od rozciągania/ściskania
37 sigma_normal = N / A
38 print(f"Naprężenie normalne od rozciągania/ściskania (\u03C3_n): {sigma_normal:.6f} Pa")
39
40 # Maksymalne naprężenie od zginania
41 sigma_bending = (M_max * h / 2) / I
42 print(f"Maksymalne naprężenie od zginania (\u03C3_b): {sigma_bending:.6f} Pa")
43
44 # Naprężenie całkowite w najniekorzystniejszym punkcie przekroju
45 sigma_total_top = sigma_normal - sigma_bending # Górne włókno
46 sigma_total_bottom = sigma_normal + sigma_bending # Dolne włókno
47
48 print("\n--- Wyniki ---")
49 print(f"Naprężenie całkowite w górnym włóknie: {sigma_total_top:.6f} Pa")
50 print(f"Naprężenie całkowite w dolnym włóknie: {sigma_total_bottom:.6f} Pa")
51

```

Rysunek 2 – Kod programu, który oblicza maksymalny moment gnący i naprężenia normalne

Ostatnim etapem było stworzenie wykresu wartości momentu gnącego na długości belki, której wartość została zdefiniowana wcześniej.

```

52 # Rysowanie wykresu momentu gnącego na długości belki
53 x = np.linspace(0, L, 500)
54 M = M + T * x
55
56 plt.figure(figsize=(10, 6))
57 plt.plot(x, M, label="Moment gnący M(x)", color="blue")
58 plt.xlabel("Długość belki (x) [m]")
59 plt.ylabel("Moment gnący (M) [Nm]")
60 plt.title("Wykres momentu gnącego na długości belki")
61 plt.grid()
62 plt.legend()
63 plt.show()
64 plt.savefig("wykres.png")
65
66 except ValueError as e:
67     print(f"Błąd: {e}")
68 except Exception as e:
69     print(f"Nieoczekiwany błąd: {e}")
70
71 if __name__ == "__main__":
72     calculate_stresses()
73

```

Rysunek 3 – Kod programu umożliwiający stworzenie wykresu zależności wartości momentu gnącego na długości belki

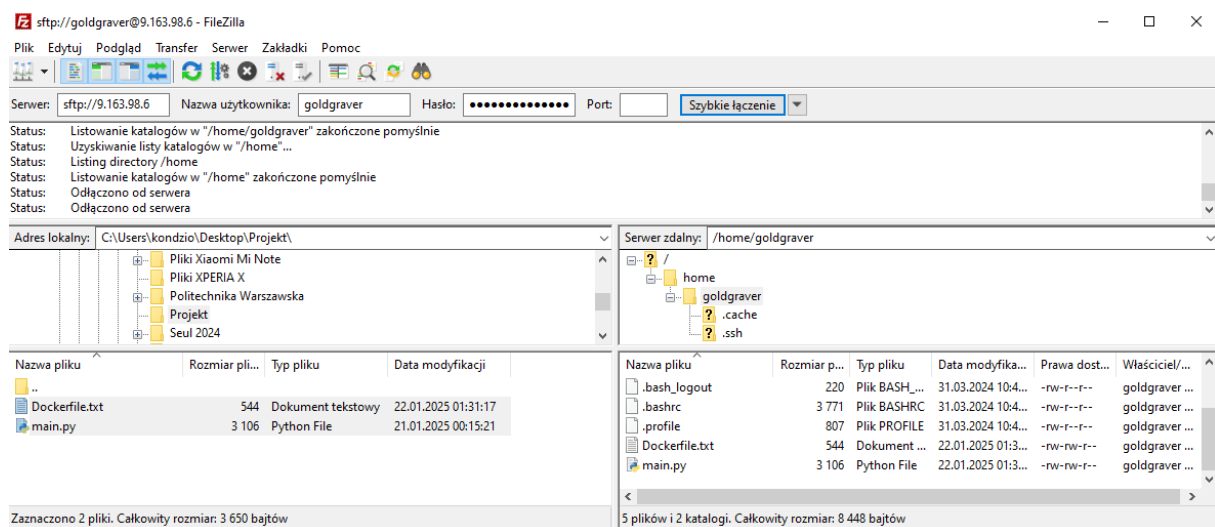
3. Wykorzystanie platformy Azure

Na platformie Azure stworzona została maszyna wirtualna z systemem operacyjnym *Linux* w wersji *Ubuntu 24.04*. Podstawowe jej elementy umieszczone zostały poniżej.

Grupa zasobów (przenieś) :	Belka_group	System operacyjny :	Linux (ubuntu 24.04)
Stan :	Uruchomione	Rozmiar :	Standard B1s (1 vcpu, 1 GiB pamięci)
Lokalizacja :	West Europe (Strefa 1)	Publiczny adres IP :	9.163.98.6
Subskrypcja (przenieś) :	Subskrypcja platformy Azure 1	Sieć/podsieć wirtualna :	Belka-vnet/default
Identyfikator subskrypcji :	ac28e445-9b28-4369-9d83-6659c38a3f84	Nazwa DNS :	Nieskonfigurowano
Strefa dostępności :	1	Stan kondycji :	-

Rysunek 4 – Podstawowe elementy maszyny wirtualnej

Następnym krokiem było połączenie się z maszyną wirtualną za pomocą aplikacji *FileZilla*, która umożliwiła skopiowanie plików *main.py* oraz *Dockerfile.txt* na maszynę wirtualną. Poniżej umieszczony został zrzut ekranu z programu *FileZilla*.



Rysunek 5 – Skopiowanie plików na maszynę wirtualną

Następnie za pomocą aplikacji *PuTTY* zainstalowany został *Docker* na maszynie wirtualnej i zbudowany obraz o nazwie *obliczenia*. W tym celu wykorzystane zostały poniższe komendy:

```
sudo apt-get update
```

```
sudo apt install docker.io
```

```
mv Dockerfile.txt Dockerfile
```

```
sudo docker build -t obliczenia .
```

```
sudo docker run --name obliczenia -it obliczenia
```

```
sudo docker ps -a
```

4. Wyniki

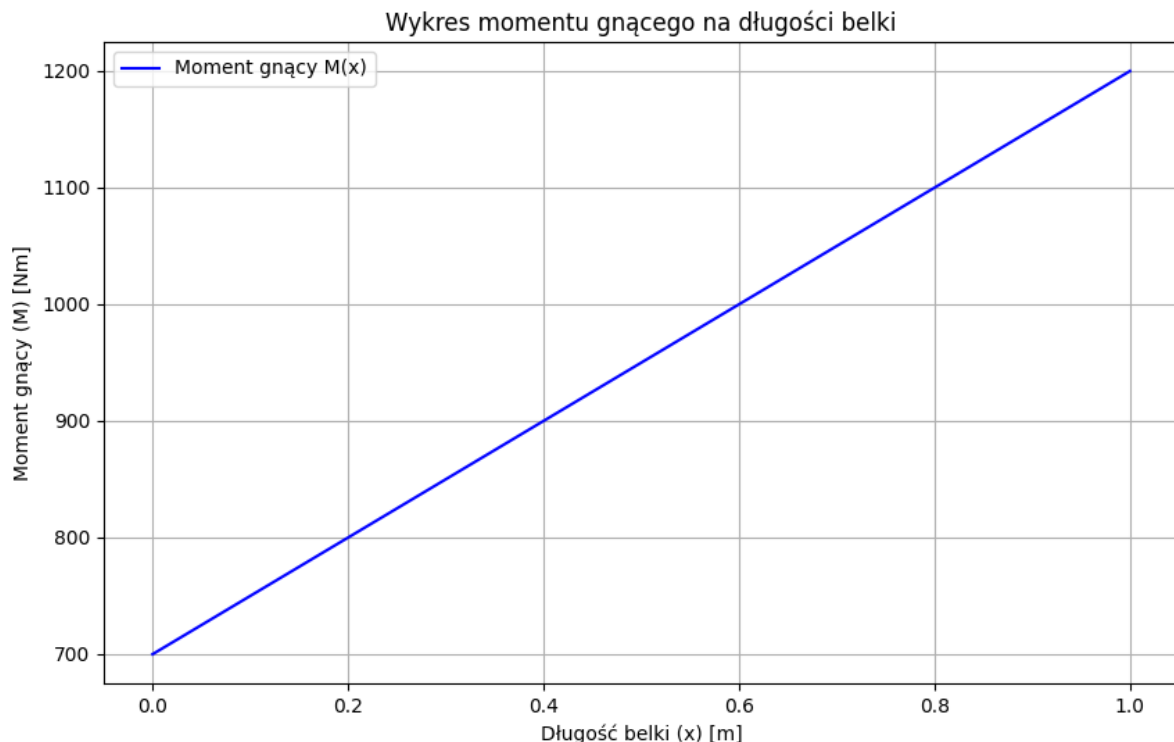
W celu sprawdzenia poprawności obliczeń wprowadzono pewne wartości danych geometrycznych obiektu, działających na niego sił oraz momentu. Na tej podstawie wyznaczone zostały wartości naprężeń normalnych oraz stworzony został wykres wartości momentu gnącego na długości belki. Wprowadzone dane i wyniki umieszczone zostały poniżej.

```
Obliczanie naprężeń normalnych od rozciągania i zginania belki prostokątnej.
Podaj szerokość przekroju prostokątnego (b) [m]: 0.01
Podaj wysokość przekroju prostokątnego (h) [m]: 0.02
Podaj długość belki (L) [m]: 1
Podaj wartość siły normalnej na swobodnym końcu belki (N) [N]: 1000
Podaj wartość siły poprzecznej na swobodnym końcu belki (T) [N]: 500
Podaj wartość momentu gnącego na swobodnym końcu belki (M) [Nm]: 700

Moment gnący w utwierdzeniu belki (M_max): 1200.000000 Nm
Pole przekroju (A): 0.000200 m^2
Moment bezwładności przekroju (I): 0.000000 m^4
Naprężenie normalne od rozciągania/ściskania ( $\sigma_n$ ): 5000000.000000 Pa
Maksymalne naprężenie od zginania ( $\sigma_b$ ): 1800000000.000000 Pa

--- Wyniki ---
Naprężenie całkowite w górnym włóknie: -1795000000.000000 Pa
Naprężenie całkowite w dolnym włóknie: 1805000000.000000 Pa
```

Rysunek 6 – Wprowadzone dane i wartości naprężeń normalnych



Rysunek 7 – Wykres wartości momentu gnącego na długości belki

5. Podsumowanie

Program zwraca poprawne w wyniki w kilku przykładowych obliczeniach, co świadczy o jego poprawnym działaniu. Umożliwia on szybkie wyznaczenie wartości maksymalnych naprężeń normalnych czy momentu gnącego, w wyniku czego analizy wytrzymałościowe prostych układów zajmą zdecydowanie mniej czasu.

Jest to prosty program, który można by rozwinąć w celu zwiększenia jego użyteczności. Wprowadzenie układu trójwymiarowego lub możliwości określenia punktu przyłożenia siły czy momentu w zdecydowany sposób zwiększyłoby jego funkcjonalność.