Obliczenia naprężeń w zbiorniku walcowym z zakończeniami kopułowymi-półsferycznymi (wklęsły i wypukły) z wykorzystaniem maszyny wirtualnej Microsoft Azure.

Kamil Zawadzki

Warszawa, 2024r.

1. Założenia zadania

Podjęte zagadnienie obliczeniowe to wyznaczenie naprężeń błonowych w ścianach zbiornika ciśnieniowego, o kształcie walcowym z zamknięciami sferycznymi, z jednej strony z wypukłym, z drugiej z wklęsłym. Obliczenia wykonano dla szerokiego zakresu wartości obciążeń ciśnieniem i rozmiarów oraz grubości zbiornika, uzyskując chmurę punktów obliczeniowych.

Geometria:

- odcinek walcowy;
- zamknięcie półsferyczne z płynnym przejściem (nieciągłość II rodzaju);
- zamknięcie półsferyczne z ostrym przejściem (wklęsłe dno) (nieciągłość I rodzaju);

Rozmiary:

Średnica wewnętrzna zmienna w zakresie *30mm – 50mm*, z rozdzielczością co *2mm;*

Grubość powłoki zmienna w zakresie *1mm – 5mm* z rozdzielczością co *1mm*;

Obciążenia:

Nadciśnienie we wnętrzu zbiornika zmienne w zakresie *3bar – 15bar,* z rozdzielczością *1bar*;

Rezultaty:

Wypisane naprężenia występujące:

- na odcinku walcowym;
- w zamknięciu sferycznym wypukłym;
- w zamknięciu sferycznym wklęsłym;
- w miejscach nieciągłości:
  - I rodzaju przy przejściu walec-półsfera wypukła;
  - II rodzaju przy przejściu walec-półsfera wklęsła;

2. Obliczenia wytrzymałościowe

2.1. Zbiornik walcowy

Naprężenia błonowe:

Naprężenia obwodowe (wzdłuż równoleżnika): $\sigma_t = \frac{p}{\delta}\rho_t = \frac{p}{\delta}R$

Naprężenia południkowe (wzdłuż tworzącej): $\sigma_p = \frac{pR}{2\delta}$

2.2. Półsferyczne zamknięcie wypukłe

Naprężenia błonowe:

Naprężenia obwodowe, południkowe: $\sigma_p = \sigma_t = \frac{pR}{2\delta}$

2.3. Półsferyczne zamknięcie wklęsłe

Naprężenia błonowe:

Naprężenia obwodowe, południkowe: $\sigma_p = \sigma_t = -\frac{pR}{2\delta}$

(po prostu ściskane zamiast rozciągania)

2.4. Program w python

Program *calc_stress.py* składa się z 3 kolejnych pętli *for*, kolejno ze względu na kolejno ciśnienie obciążenia, grubość ścianki i średnicę cylindra. Program zapisuje wyniki w kolejnych wierszach pliku *results.txt*, oraz wyświetla w terminalu.

3. Obliczenia lokalne w środowisku Python

Przygotowano kod w środowisku python 3.11 realizujący obliczenia omówione wyżej dla szeregu warunków obliczeń o których także wspomniano wyżej. Po przygotowaniu obrazu dockera, uruchomiono obraz i wykonano obliczenia.

W pierwszej kolejności utworzono obraz dockera *tank_stress_calculator*.

```
D:\Uczelniane\mgr\sem4\chmurka\projekt\python>docker build -t tank_stress_calculator:1 .
[+] Building 21.1s (13/13) FINISHED                                                                    docker:default
 => [internal] load build definition from Dockerfile                                                           0.0s
 => => transferring dockerfile: 224B                                                                           0.0s
 => [internal] load .dockerignore                                                                              0.0s
 => => transferring context: 2B                                                                                0.0s
 => [internal] load metadata for docker.io/library/python:3.11                                                 2.3s
 => [auth] library/python:pull token for registry-1.docker.io                                                  0.0s
 => [1/7] FROM docker.io/library/python:3.11@sha256:a2f76e7c73c241d15e867987f143dfbdcf280fb229ae1ecde98850504fd3b  1.2s
 => => resolve docker.io/library/python:3.11@sha256:a2f76e7c73c241d15e867987f143dfbdcf280fb229ae1ecde98850504fd3b  0.0s
 => => sha256:a2f76e7c73c241d15e867987f143dfbdcf280fb229ae1ecde98850504fd3b234 2.14kB / 2.14kB                 0.0s
 => => sha256:396abfa55bae5d59681d756c99ebba84c7e8d9bd2efea15b6dc5c623ba04648f 2.01kB / 2.01kB                 0.0s
 => => sha256:52f488aeec1e04055ecb54510891a2103f38d694c6ccff612589ea83d3ea7d04 7.34kB / 7.34kB                 0.0s
 => => sha256:31c0f7a3374e7ab700f2667ffe97a8ddd8017d1c00de9489bbc05523fd5d4d93 3.11MB / 3.11MB                 0.5s
 => => extracting sha256:31c0f7a3374e7ab700f2667ffe97a8ddd8017d1c00de9489bbc05523fd5d4d93                      0.5s
 => [internal] load build context                                                                              0.0s
 => => transferring context: 4.95kB                                                                            0.0s
 => [2/7] WORKDIR /app                                                                                         0.1s
 => [3/7] COPY requirements.txt .                                                                              0.1s
 => [4/7] RUN apt-get update                                                                                   5.0s
 => [5/7] RUN apt-get install vim -y                                                                          11.1s
 => [6/7] RUN mkdir -p /home/cloud                                                                             0.6s
 => [7/7] COPY . /app                                                                                          0.1s
 => exporting to image                                                                                         0.5s
 => => exporting layers                                                                                        0.4s
 => => writing image sha256:683bf99541e687f710b352dd66df35499df51f78dac909881ddc05f00e81ac41                   0.0s
 => => naming to docker.io/library/tank_stress_calculator:1                                                    0.0s

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview
```

Po sprawdzeniu poprawności utworzenia obrazu dockera i jego ID:

```
D:\Uczelniane\mgr\sem4\chmurka\projekt\python>docker images
REPOSITORY                TAG       IMAGE ID       CREATED             SIZE
tank_stress_calculator    1         683bf99541e6   About a minute ago   1.07GB
<none>                    <none>    5b0d2e076306   2 days ago           1.07GB
tank_stress_calc          2         d5eaa0378be9   2 days ago           1.01GB
naprezenia                1         d14a4e519f91   4 days ago           1.07GB
calc_stress               latest    cb036f2bcf97   8 days ago           1.16GB
calc_stress               tag       937339ff0a09   10 days ago          1.16GB
docker/welcome-to-docker  latest    c1f619b6477e   2 months ago         18.6MB
python39                  latest    c9413de1b43d   2 months ago         1.06GB
```

Można było uruchomić obraz:

```
D:\Uczelniane\mgr\sem4\chmurka\projekt\python>docker run tank_stress_calculator:1
```

Wywołany kod potrzebował 0,5128 sekundy do ukończenia obliczeń lokalnie.

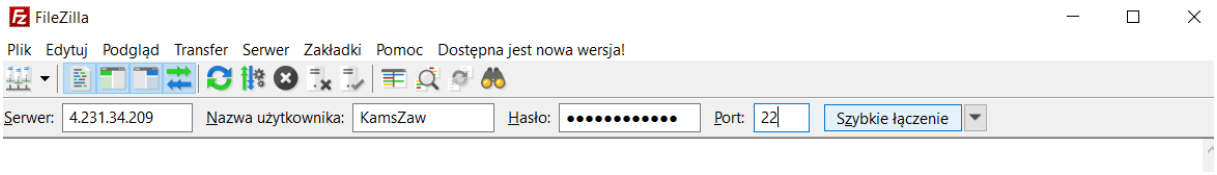
```
=====================================
Pressure: 15 bar, Thickness: 5 mm, Diameter: 100 mm
Von Mises Stress in Cylinder: 202.36 MPa
Von Mises Stress in Dome: 95.46 MPa
Von Mises Stress in Dome wkl: 95.46 MPa
=====================================
Run time: 0.5128 sec
```

4. Przygotowanie środowiska wirtualnego Azure

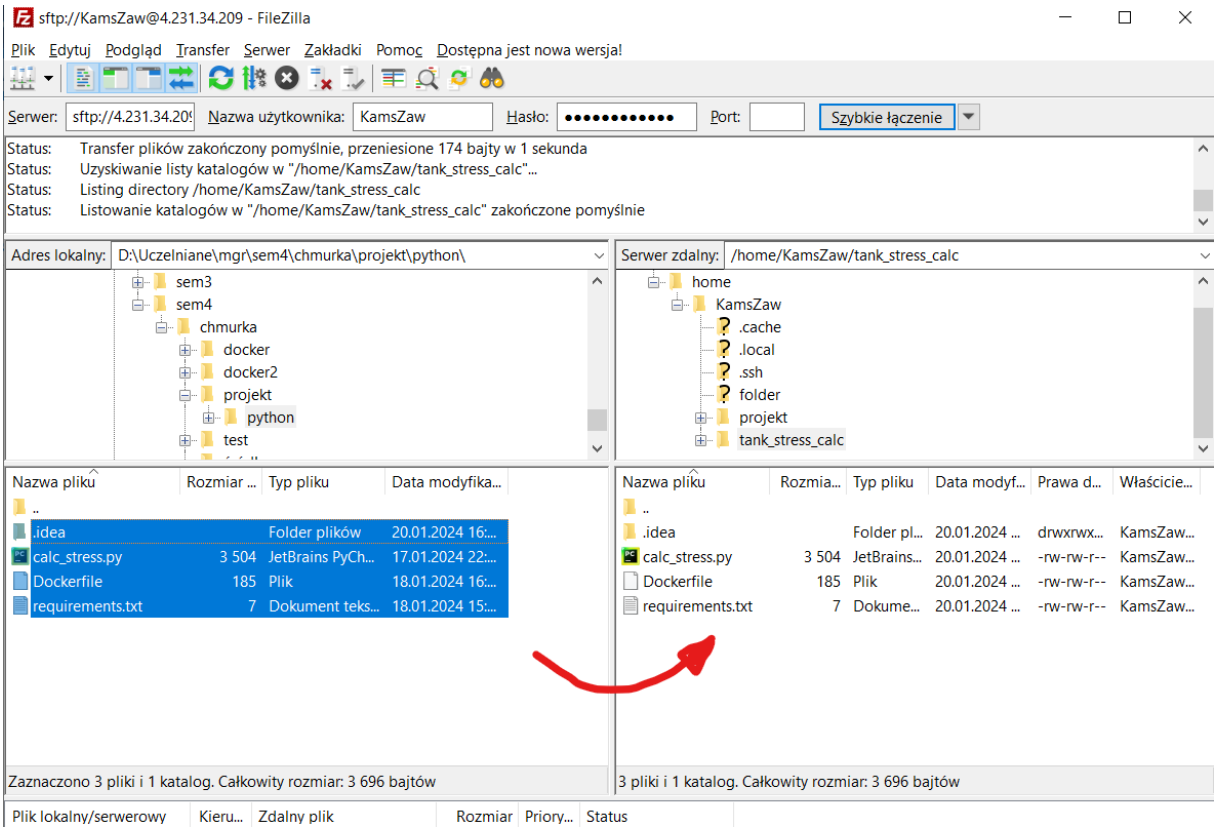
Na platformie Microsoft Azure utworzono maszynę wirtualną działającą na systemie operacyjnym Ubuntu 20.04, wyposażoną w 2 procesory wirtualne vcpu i 4 GB pamięci. Dodatkowo wyposażona jest w podpięty dysk pamięci 30GB, pozwalający na przechowywanie niezbędnych plików.

W celu umożliwienia przeprowadzenia obliczeń na maszynie wirtualnej konieczne było przekazanie plików koniecznych do uruchomienia programu na maszynie wirtualnej, do czego przydatna była aplikacja *Filezilla*, umożliwiająca zarządzanie w przejrzystym interfejsie graficznym plikami zarówno lokalnymi, jak i na maszynie wirtualnej.

Pierwszym krokiem było zalogowanie się do maszyny wirtualnej z poziomu *Filezilla*:



Na maszynie wirtualnej utworzono nowy folder w ścieżce */home/KamsZaw/tank_stress_calc*, a następnie przekopiowano pliki z folderu lokalnego do maszyny wirtualnej:



5. Obliczenia w środowisku wirtualnym

Uruchomienie obliczeń na maszynie wirtualnej odbywa się analogicznie jak na komputerze lokalnym. Po wejściu do folderu zawierającego wszystkie potrzebne pliki możliwe było utworzenie obrazu dockera, a następnie jego uruchomienie.

Najpierw po zalogowaniu do maszyny wirtualnej otwarto lokalizację ścieżki */home/KamsZaw/tank_stress_calc*, a następnie utworzono obraz dockera komendą:

```
KamsZaw@Virtualtutorial:~$ cd /home/KamsZaw/tank_stress_calc
KamsZaw@Virtualtutorial:~/tank_stress_calc$ docker build -t tank_stress_calculating:1 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  19.46kB
Step 1/8 : FROM python:3.11
 ---> 1c906257ce07
Step 2/8 : WORKDIR /app
 ---> Using cache
 ---> 01a0151a4f36
Step 3/8 : COPY requirements.txt .
 ---> Using cache
 ---> ae3861cd9668
Step 4/8 : RUN apt-get update
 ---> Using cache
 ---> 52fe1a69890a
Step 5/8 : RUN apt-get install vim -y
 ---> Using cache
 ---> ecb897897e56
Step 6/8 : RUN mkdir -p /home/cloud
 ---> Using cache
 ---> 2ab5fc4e4dd0
Step 7/8 : COPY . /app
 ---> Using cache
 ---> a57efdb4698f
Step 8/8 : CMD ["python","calc_stress.py"]
 ---> Using cache
 ---> 1c3c77aad8ba
Successfully built 1c3c77aad8ba
Successfully tagged tank_stress_calculating:1
KamsZaw@Virtualtutorial:~/tank_stress_calc$
```

Następnie sprawdzono poprawność utworzenia obrazu dockera i jego ID:

```
KamsZaw@Virtualtutorial:~/tank_stress_calc$ docker images
REPOSITORY                TAG       IMAGE ID       CREATED       SIZE
tank_stress_calculating   1         1c3c77aad8ba   2 days ago    1.07GB
naprezenia                1         d14a4e519f91   4 days ago    1.07GB
calc_stress               latest    cb036f2bcf97   8 days ago    1.16GB
calc_stress               tag       937339ff0a09   10 days ago   1.16GB
python                    3.11      1c906257ce07   6 weeks ago   1.01GB
KamsZaw@Virtualtutorial:~/tank_stress_calc$ docker run tank_stress_calculating:1
```

Wywołany obraz potrzebował $0{,}2264$ sekundy na ukończenie obliczeń na maszynie wirtualnej, co oznacza zysk 56% czasu.

```
========================================
Pressure: 15 bar, Thickness: 5 mm, Diameter: 100 mm
Von Mises Stress in Cylinder: 202.36 MPa
Von Mises Stress in Dome: 95.46 MPa
Von Mises Stress in Dome wkl: 95.46 MPa
========================================
Run time: 0.2264 sec
KamsZaw@Virtualtutorial:~/tank_stress_calc$
```

6. Podsumowanie

Wykorzystanie maszyny wirtualnej pozwala na odciążenie maszyny lokalnej z obliczeń, co ma szczególne znaczenie przy bardziej wymagających obliczeniach. Możliwe jest skonfigurowanie maszyny wirtualnej o podzespołach bardziej wydajnych od posiadanego komputera osobistego, bez konieczności ich zakupu, a jedynie „wynajmując” zasoby.