Metody Elementów Skończonych

Sprawozdanie wykonał **Mikita Shmialiou**

Temat: Symulacja ustalonych & nieustalonych procesów cieplnych

**Symulacja ustalonych procesów cieplnych**

*Wstęp*

Celem tej części sprawozdania jest obliczenie rozkładu temperatury w jednowymiarowym pręcie. Zadanie będzie rozwiązane na kilka sposobów:

1. Analityczne rozwiązanie układu równań oraz napisanie kodu w Pythonie na podstawie tego rozwiązania
2. przez bezpośrednią minimalizację funkcjonału w programie Excel

*Przedstawienie modelu z warunkami brzegowymi*

Analiza odbywa się na pręcie o długości L przy procesie ustalonego przewodnictwa ciepła. Zakładamy, że wymiana ciepła odbywa się tylko na końcach pręta:

1. na początku pręta mamy strumień ciepła q
2. na końcu mamy konwekcję

A blue rectangular object with a blue rectangle

AI-generated content may be incorrect.

*Podstawy rozwiązania MES dla problemu optymalizacji bezpośredniej*

Rozwiązanie polega na poszukiwaniu minimum funkcjonału energetycznego J. Przy jednowymiarowym, ustalonym przepływie ciepła oraz z warunkami brzegowymi J wygląda:

Dane wejściowe:

; ;;

***Wyniki MES dla problemu rozwiązywanego Excelem***

*3-węzlowy element*

A number and numbers on a white background

AI-generated content may be incorrect.

A graph on a sheet of paper

AI-generated content may be incorrect.

Wyniki w EXCELu są bardzo bliskie do tych, co były otrzymane analitycznie, co świadczy o tym, że znalezienie temperatur za pomocą SOLVER’a (minizacja) odbyło się poprawnie.

*5-węzlowy element*

Dla obliczenia 5 węzłów trzeba delikatnie zmodyfikować plik EXEL’owy. Żeby osiągnąć ten cel trzeba podzielić nasz pręt o długość L na 4 elementy skończony. W rezultacie ilość węzłów będzie wynosiła 5. Trzeba rozszerzyć poprzednie równania do 5 węzłów.

Dane wejściowe, które uległy zmianie:

Globalna macierz sztywności dla 5 węzłów (5x5):

A group of black letters

AI-generated content may be incorrect.

[alfaS w ostatnim węźle to konwekcja]

A number with numbers on it

AI-generated content may be incorrect.

Wektor obciążeń:

A black rectangular object with numbers and lines

AI-generated content may be incorrect.

Ostateczny układ równań dla 5 węzłów:

A number lines and numbers

AI-generated content may be incorrect.

Z tego równania wynika, że temperatury są równe:

Wzór J dla 5-węzłowego układu:

A graph on a sheet of paper

AI-generated content may be incorrect.

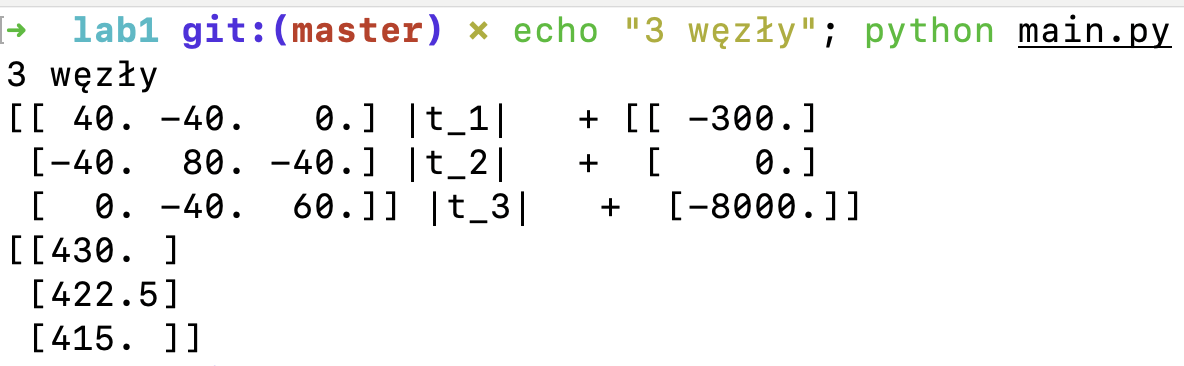
Wyniki w EXCELu są bardzo bliskie do tych, co były otrzymane analitycznie, co świadczy o tym, że znalezienie temperatur za pomocą SOLVER’a (minizacja) odbyło się poprawnie.

*Porównanie z kodem*

Kod:



(Ważne – zmiana zmiennej dir nie jest poprawnie uwzględniona, czyli wyniki będę złe)

Wyniki odpalania programu dla 3 i 5 węzłów A screenshot of a computer code

AI-generated content may be incorrect.

Jak widać wyniki są identycznie wynikom otrzymanym analitycznie

Porównanie wyników:

3 węzły

|  |  |  |  |
| --- | --- | --- | --- |
|  | Kod | Excel |  |
|  | 430 | 429.9999989 | 1.14895E-06 |
|  | 422.5 | 422.4999732 | 2.67601E-05 |
|  | 415 | 415.0000174 | -1.7359E-05 |

5 węzłów

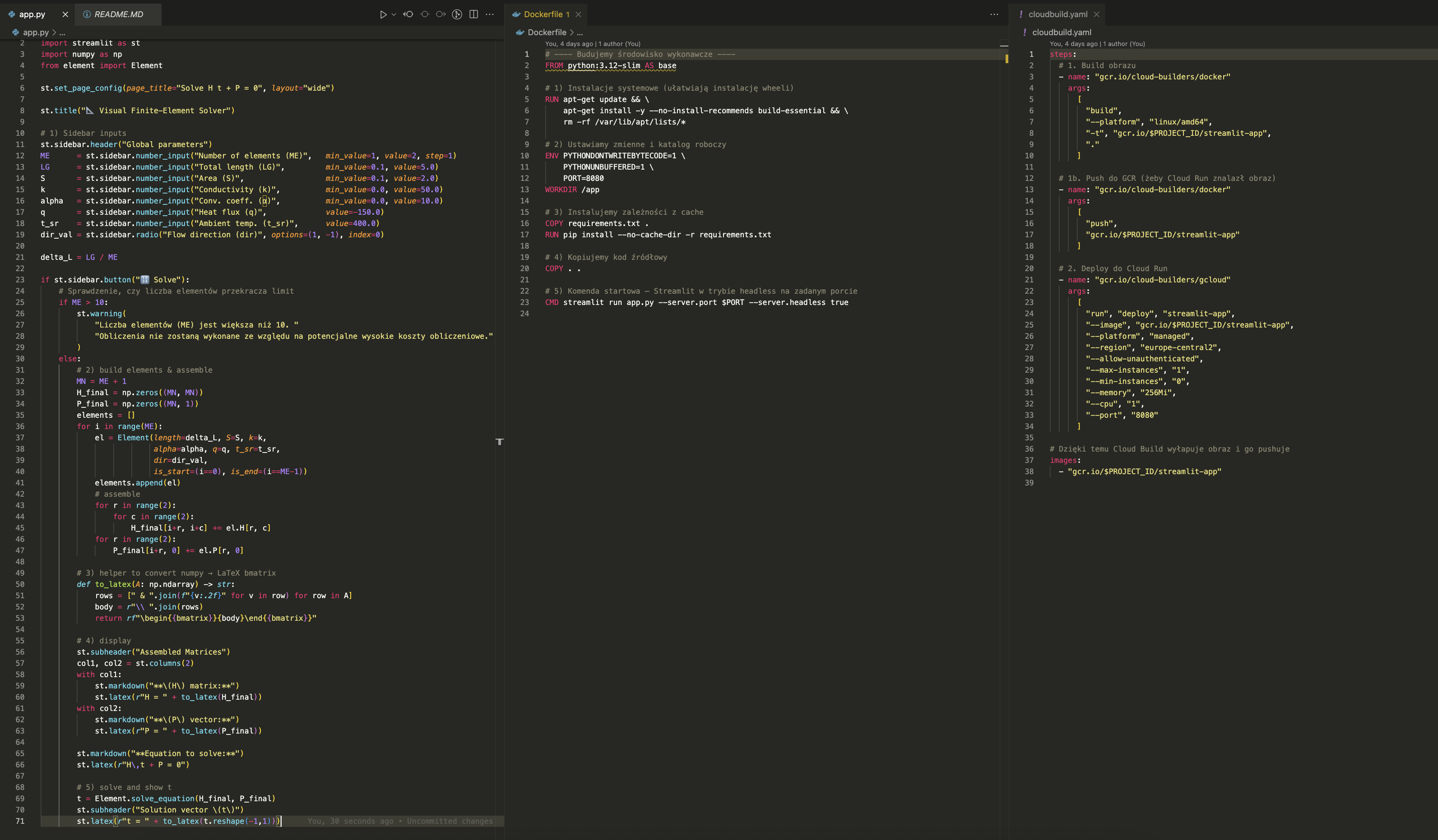
|  |  |  |  |
| --- | --- | --- | --- |
|  | Kod | Excel |  |
|  | 430 | 429.9892231 | 1.077688E-02 |
|  | 426.25 | 426.2394368 | 1.056325E-02 |
|  | 422.5 | 422.4900778 | 9.922192E-03 |
|  | 418.74 | 418.7411473 | -1.147340E-03 |
|  | 415 | 414.9926552 | 7.344792E-03 |

Różnice między wynikami z kodu i EXCEL’a są niewielkie i wynikają z dokładności narzędzia SOLVER.

**Code Addon**

Po sprawdzaniu poprawności działania kodu dla obliczenia ustalonego procesu powstał pomysł o zrobieniu wizualizacji do kodu. Wizualizacja polega na korzystaniu z dodatkowych bibliotek, żeby stworzyć graficzny widok do zaprezentowania wyników oraz ułatwienia korzystania z stworzonego narzędzia.

Kod:



Ten kod był wygenerowany za pomocą ChatGPT na podstawie kodu, który był umieszczony wcześniej i służy wyłącznie do wizualizacji.

„Projekt” składa się z 3 plików:

1. app.py – wizualizacja + logika z pliku main.py
2. Dockerfile
3. Cloudbuild.yaml

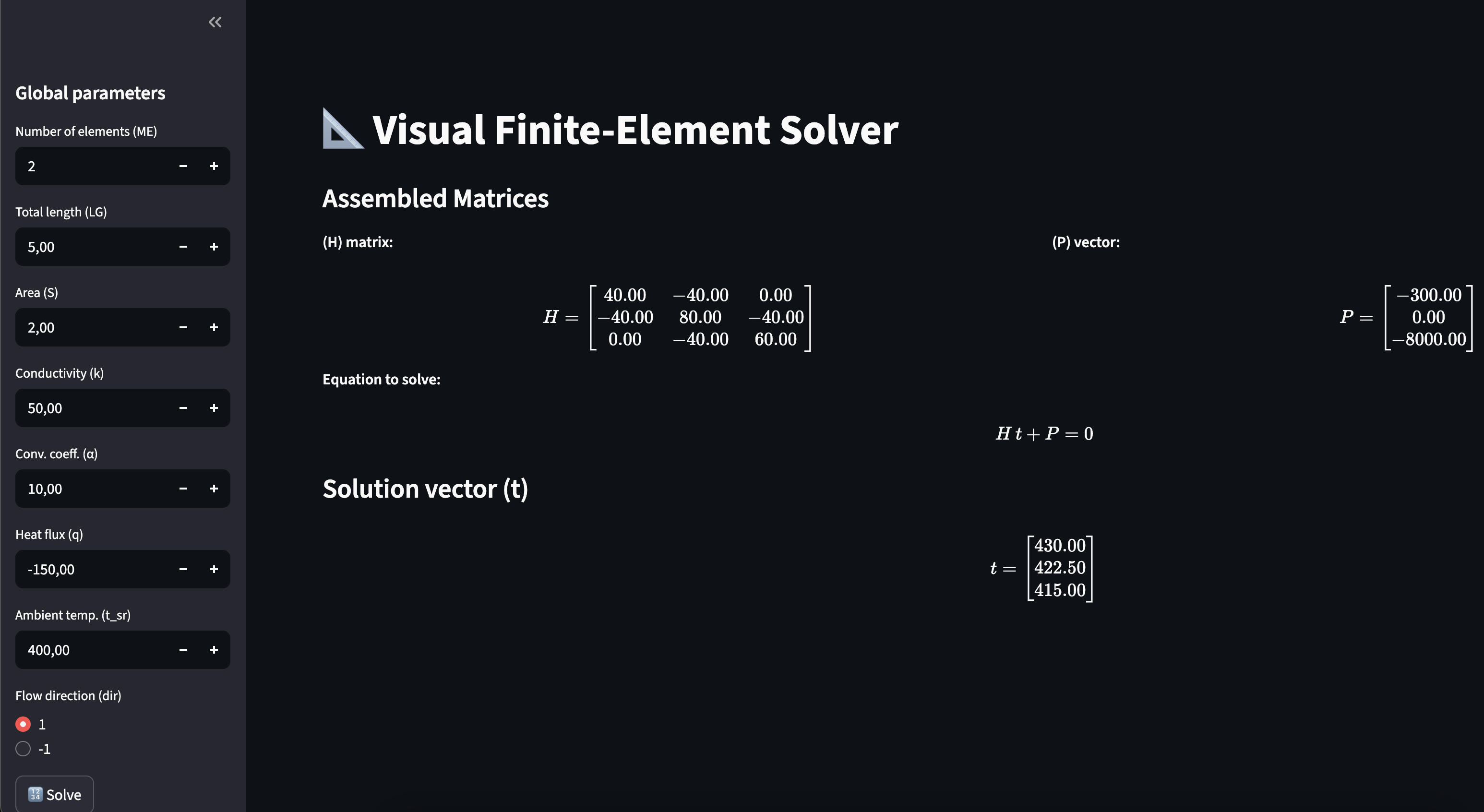
Pliki Dockerfile oraz Cloudbuild.yaml są potrzebne dla możliwości umieszczenia tego projektu w chmurze GCP, żeby dodać możliwość zaprezentowania tego programu wielu ludziom.

Aplikacja będzie dostępna w chmurze pod następnym linkiem (do 29/06/2025):

<https://streamlit-app-83611701832.europe-central2.run.app/>

**Demo**

2 elementy



4 elementy

A screenshot of a computer

AI-generated content may be incorrect.

10 elementów

A screenshot of a computer

AI-generated content may be incorrect.

*Podsumowanie i wnioski*

W ramach sprawozdania była zrobiona symulacja ustalonego przepływu ciepła w pręcie za pomocą MES. Zastosowane rozwiązania:

1. Rozwiązanie analityczne (układ równań)
2. Minimalizacja funkcjonału za pomocą SOLVER’a
3. Rozwiązanie własnym programem w Pythonie

Wszystkie metody dały zbliżone wyniki, co potwierdza poprawność modeli. Zwiększenie liczby węzłów daje nam bardziej szczególny rozkład temperatury, trzymając się przy tym zgodności wartości na brzegach. Stworzenie aplikacji webowej nie było wymagane, ale to stanowi praktyczne rozszerzenie „projektu” oraz ułatwia prezentację oraz analizę wyników.

**Symulacja nieustalonych procesów cieplnych**

*Wstęp*

Celem tej części sprawozdania jest analiza niestacjonarnego (zmiennego w czasie) procesu przepływu ciepła we wsadzie o przekroju okrągłym. Analiza będzie zrobiona będzie zrobiona za pomocą programu w Pythonie (który był napisany na podstawie programu w Fortranie), który używa MES do rozwiązania problemu.

Program należy odpalić z różnymi parametrami:

1. Różna ilość elementów bez zmiany innych parametrów i odpowiedzieć na pytanie dla jakiej ilości węzłów rozwiązanie jest stabilne.
2. Różna długość kroku czasowego bez zmiany innych parametrów i wywnioskować na co wpływa długość kroku czasowego.

*Przedstawienie modelu z warunkami brzegowymi*

Analizowano proces nieustalonego przewodnictwa ciepła we wsadzie o przekroju okrągłym (rys. 6.1). Założono, że wymiana ciepła będzie odbywała się w sposób osiowo-symetryczny (rys. 6.2).

A black and white image

AI-generated content may be incorrect.

Rys. 6.1. Przykładowy rozkład temperatury we wsadzie o przekroju okrągłym

A circle with a line in the center

AI-generated content may be incorrect.

Rys. 6.2. Schemat obliczeniowy do wyznaczenia nieustalonego rozkładu temperatury

*Podstawy rozwiązania MES dla problemu niestacjonarnego*

Podstawą analizy jest równanie Fouriera dla procesu niestacjonarnego

Zgrubna charakterystyka kodu

Analiza była zrobiona na podstawie kodu w Pythonem, który był napisany na podstawie kodu w Fortranie (TEMP1D). Program używa bibliotekę csv do zapisu wyników do pliku oraz bibliotekę matplotlib do rysowania wykresów.

Poniżej umieszczam najbardziej istotną część kodu:

A screen shot of a computer program

AI-generated content may be incorrect.

Kod głównie polega na tych wzorach:

A group of mathematical equations

AI-generated content may be incorrect.

*Analiza wpływu wielkości elementu oraz długości kroku czasowego na wyniki obliczeń*

1. Parametry standardowe

dt = 0.4367 s

Czas: 2.498515 s

A graph on a white sheet

AI-generated content may be incorrect.

Jak widać temperatury się zbliżają, co jest oczekiwane. Wsad się grzeje przez temperaturę otoczenia i dlatego pomarańczowy wykres jest opóźniony – ma mniejszą temperaturę niż powierzchnia, ale po pewnym czasie temperatury powinny się wyrównać.

A screen shot of a graph

AI-generated content may be incorrect.

Wyniki oraz dodatkowy wykres różnicy temperatury pokazują, że po 1000 sekundach różnica wynosi ~19 stopni.

2. Zmiana liczy elementów

|  |  |  |
| --- | --- | --- |
| NE | time, s | dt, s |
| 50 | 2.17050815 | 0.4367 |
| 100 | 13.1513889 | 0.1092 |
| 150 | 69.3653231 | 0.0485 |
| 200 | 252.294886 | 0.0273 |
| 250 | 473.519242 | 0.0175 |
| 300 | 795.241153 | 0.0121 |
| 350 | 1222.50325 | 0.0089 |
| 400 | 1660.02613 | 0.0068 |
| 450 | 2113.89599 | 0.0054 |
| 500 | 2534.29775 | 0.0044 |

Zależność czasu wykonania programu od ilości elementów

Zależność dt od ilości elementów

*Analiza wyników*

Na podstawie danych możemy wywnioskować, że zwiększenie liczby elementów bezpośrednio wpływa na:

1. czas obliczeń
   1. jest większy, bo zwiększa się liczba kroków
   2. więcej obliczeń w każdym kroku
2. długość dt (wzór na obliczenie dt używa liczbę elementów)
3. precyzję obliczeń

Możemy zauważyć, że wykres dt(NE) się wypłaszacza i dalsze zwiększenie liczby elementów niewiele zwiększy precyzję, ale znacznie zwiększy koszty obliczeniowe. Na szybki rzut oka wygląda, że złożoność kodu jest – wyniki czasowe mogą z tym się zgadzać, bo wszystkie obliczenia (dla każdej ilości elementów) było odpalone w tym samym czasie (żeby szybciej policzyć) co mogło spowodować wolniejszą pracę każdego z tych procesów (każde liczba wątków – to osobny proces, który w sobie zawierał wątki).

Również można zauważyć, że wyniki przy 200 i 500 są prawie identyczne, ale liczba wierszy jest o ~6.3 większa. Czyli jeśli rozwiązanie nie wymaga bardzo dużej precyzji, to można się ograniczyć 200 elementami.

A screenshot of a computer

AI-generated content may be incorrect.

3. Zmiana długości kroku czasowego

Zmiana dTau się odbywa przez modyfikację następnej linijki

dTau = dR \*\* 2 / (0.5 \* a)

Parametry standardowe:

dt = 0.4367 s

Czas: 7.677455s

dT na końcu obliczeń: 20.0313

A graph with a line

AI-generated content may be incorrect.A graph with a line

AI-generated content may be incorrect.

|  |  |  |  |
| --- | --- | --- | --- |
|  | dt, s | time, t | dT, K |
| dt/8 | 0.0546 | 68.983486 | 19.9893 |
| dt/4 | 0.1092 | 37.500213 | 19.9953 |
| dt | 0.4367 | 7.677455 | 20.0313 |
| 4dt | 1.7452 | 1.769638 | 20.1752 |
| 8dt | 3.4843 | 0.85683 | 20.3668 |
| 100dt | 43.4783 | 0.067629 | 24.8586 |

*Analiza wyników*

**Na co wpływa długość kroku czasowego?** Długość kroku czasowego wpływa na czas obliczeń (kolumna *time*) oraz powinna wpływać na precyzję obliczeń. Chociaż różnica pomiędzy końcową temperaturą przy nie jest dużo i wynosi:

Widoczną różnice widać przy 100dt i ona wynosi 4.8273K. Ale dalej widać czy wyniki końcowe są podobne. Ta długość przede wszystkim wpływa na precyzję obliczeń przy używaniu metody Gausa.