Wydział Informatyki i Sztucznej Inteligencji

LAB1a: Budowa modelu bazowego "do przodu" - implementacja algorytmu dla jednowymiarowego, stacjonarnego przypadku transferu ciepła w materiale jednorondym

dr inż. Konrad M. Gruszka,*

4 marca 2025

Streszczenie

Ten dokument stanowi instrukcję do wykonania ćwiczenia z zagadnienia transferu ciepła w jednowymiarowym, stacjonarnym przypadku do przedmiotu Rozwiązywanie Zadań Odwrotnych. Bazując na niniejszym opisie należy zaprojektować i zaimplementować algorytm MRS w oparciu o kryteria przedstawione w dalszej części tego dokumentu.

1 Metoda różnic skończonych - MRS

Wprowadzenie
Wraz z tym dokumentem otrzymali Państwo również załącznik: opis metody MRS w przypadku jednowymiarowym dla stajonarnego problemu transferu ciepła. Należy dokładnie zapoznać się z tym dokumentem, gdyż stanowi on bazę matematyczną, konieczną do rozwiązania postawionego przed Państwem zadania.

Cel ćwiczenia
Celem ćwiczenia jest z**aprojektowanie i implementacja algorytmu rozwiązującego problem jednowymiarowego stacjonarnego transferu ciepła przy użyciu metody różnic skończonych** w języku Python. Efektem końcowym jest skrypt, który pozwoli na analizę rozkładu temperatury wzdłuż jednorodnego pręta.

Środowisko pracy

- Python w wersji > 3
- Visual Studio Code z dodatkiem Jupyter Notebook

Biblioteki

- numpy
- matplotlib
- time

Wymagania funkcjonalne algorytmu

1. Definicja ilości węzłów siatki: Algorytm musi pozwalać użytkownikowi na zdefiniowanie ilości węzłów siatki, co ma decydujacy wpływ na rozdzielczość a przez to dokładność symulacji.

*Katedra Informatyki, Wydział Inżynierii Mechanicznej i Informatyki (kgruszka@icis.pcz.pl)

2. Warunki brzegowe: Użytkownik musi mieć możliwość zdefiniowania temperatur na obu końcach 1-wymiarowego pręta.
3. Iteracyjne ustalanie temperatury:
   - W wersji podstawowej, algorytm zakończy iteracje po osiągnięciu zadanej ilości iteracji.
   - W wersji rozszerzonej, kryterium zakończenia iteracji ma być zbieżność rozwiązań, tj. zmiana temperatur w kolejnych iteracjach musi być poniżej zadanej tolerancji.
   - Użytkownik ma mieć możliwość wyboru kryterium zakończenia iteracji (po całkowitej liczbie iteracji lub po osiągnięciu zbieżności obliczeniowej.
4. Wizualizacja wyników: temperatury w poszczególnych węzłach powinny być wizualizowane na wykresie, na którym oś X reprezentuje numer węzła, a oś Y temperaturę (matplotlib).
5. Skrypt ma umożliwić wywołanie z linii poleceń wraz z parametrami symulacji oraz bezpośrednio z VSCode.

Wydajność obliczeniowa W celu ustalenia metryk wydajności obliczeniowej oraz ustalenia, w jaki sposób dobór parametrów empirycznych symulacji wpływa na całkowity czas wykonania należy zaimplementować:

- Pomiar czasu wykonania algorytmu: należy zmierzyć czas wykonania algorytmu od rozpoczęcia do zakończenia pojedynczej iteracji.
- Pomiar całkowitego czasu wykonania wszystkich iteracji do uzyskania zbieżności lub zakończenia po ustalonej ilości iteracji.
- Dane o wydajności: po zakończeniu iteracji należy wypisać, po ilu iteracjach osiągnięto zbieżność dla zadanej tolerancji.

Zadania do wykonania oraz do implementacji algorytmu w Pythonie

1. Przygotuj funkcję ‘simulate_heat_transfer(N, T0, TN, max_iter, tolerance=None)‘, gdzie:
   - ‘N‘ to liczba węzłów siatki,
   - ‘T0‘, ‘TN‘ to temperatury na końcach pręta,
   - ‘max_iter‘ to maksymalna liczba iteracji (dla wersji podstawowej),
   - ‘tolerance‘ to tolerancja zbieżności (dla wersji rozszerzonej).
   - Funkcja ta powinna zwracać listę temperatur (temperaturę dla każdego węzła).
2. Analiza zbieżności:
   - Wyjaśnij, co oznacza zbieżność w kontekście tego algorytmu i jak jest obliczana.
3. Testowanie i dokumentacja:
   - Przetestuj algorytm dla różnych wartości ‘N‘, ‘T_0‘, ‘T_N‘, ‘max_iter‘, i ‘tolerance‘.
   - Dokumentujcie każdy test oraz wyniki.

2 Forma i ocena wykonania ćwiczenia

Forma oddania zadania
Zadanie należy oddać w formie pliku z rozszerzeniem .py zawierającego implementację algorytmu. Należy przeprowadzić test dla następujących parametrów:

- Liczba węzłów N: 50
- $\kappa = 237.0$ [W/m·K] (aluminium)
- $T_0 = 150$
- $T_N = 50$
- $tolerancja = 0.5K$
- $max\_iter = 1000$

Ponadto należy przygotować raport (pdf), w którym znaleźć się mają następujące informacje:

– Udokumentowanie przeprowadzonego testu dla parametrów podanych powyżej, wyniki oraz odpowiednie komentarze i wyjaśnienia.

– Uzyskane czasy trwania pojedynczej iteracji i całkowitego czasu trwania wszystkich iteracji w odniesieniu do parametrów max_iter oraz dla algorytmu z tolerancją.

– Wykres przedstawiający rokład temperatur w pręcie.

Ocena ćwiczenia będzie bazować na:

1. Poprawności implementacji algorytmu.
2. Kompletności przeprowadzonych testów.
3. Jakości dokumentacji i wyjaśnień teoretycznych.
4. Efektywności i optymalizacji kodu.