

# LAB1b: Budowa modelu bazowego "do przodu" - implementacja algorytmu dla dwuwymiarowego, stacjonarnego przypadku transferu ciepła w materiale jednorodnym

dr inż. Konrad M. Gruszka,\*

4 marca 2025

## Streszczenie

Ten dokument stanowi instrukcję do wykonania ćwiczenia z zagadnienia transferu ciepła w dwuwymiarowym, 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 dwuwymiarowym dla stacjonarnego 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 zaprojektowanie i implementacja algorytmu rozwiązującego problem dwuwymiarowego stacjonarnego transferu ciepła przy użyciu metody różnic skończonych w języku Python. Praca odbywa się w małych grupach (do 5 osób) a efektem końcowym jest skrypt, który pozwoli na analizę rozkładu temperatury w dwuwymiarowym materiale jednorodnym.

### 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 w dwóch wymiarach, co ma decydujący wpływ na rozdzielczość a przez to dokładność symulacji. Zakładamy regularną implementację siatki w obydwu wymiarach.
2. Warunki brzegowe: Użytkownik musi mieć możliwość zdefiniowania temperatur na wszystkich czterech stronach 2-wymiarowej powierzchni.

---

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

## 3. Iteracyjne ustalanie temperatury:

- W wersji podstawowej, program zakończy obliczenia po osiągnięciu zadanej ilości iteracji.
  - W wersji rozszerzonej, kryterium zakończenia iteracji ma być zbieżność rozwiązań, tj. zmiany temperatur w kolejnych iteracjach poniżej zadanej tolerancji.
  - Użytkownik ma mieć możliwość wyboru kryterium zakończenia iteracji (po całkowitej liczbie iteracji lub po osiągnięciu założonej zbieżności obliczeniowej).
4. Wizualizacja wyników: temperatury w poszczególnych węzłach powinny być wizualizowane na dwuwymiarowym wykresie, na którym kolor reprezentuje otrzymaną temperaturę.
5. Skrypt ma umożliwić wywołanie z linii poleceń wraz z parametrami symulacji oraz bezpośrednio z VSCode.

**Wydaźność 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 obliczeń 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_transfer2D(Nx, Ny, TU, TD, TL, TR, max_iter, tolerance=None)`, gdzie:
- `Nx` oraz `Ny` to liczba węzłów siatki, w odpowiednio osi X i osi Y,
  - `TU`, `TD`, `TL`, `TR` to temperatury na brzegach materiału (`TU`-Up, `TD`-Down, `TL`-Left, `TR`-Right),
  - `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 w 2D).
2. Analiza zbieżności:
- Sprawdź, co powoduje zmiana tego parametru w kontekście otrzymywanych wyników.
3. Testowanie i dokumentacja:
- Przetestuj algorytm dla różnych wartości `Nx`, `Ny`, `TU`, `TD`, `TL`, `TR`, `max_iter`, i `tolerance`.
  - Dokumentuj 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  $N_x$ : 25
- Liczba węzłów  $N_y$ : 25
- $\kappa = 237.0[W/m \cdot K]$  (aluminium)
- $T_U = 300, T_D = 100$
- $T_L = 200, T_R = 0$
- $tolerancja = 0.5K$
- $max\_iter = 1000$

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 rozkład temperatur w przecie.

### Ocena ćwiczenia będzie bazować na:

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