

LAB3: Zastosowanie metod gradientowych do rozwiązania zadania odwrotnego

dr inż. Konrad M. Gruszka,*

4 marca 2025

Streszczenie

Ten dokument stanowi instrukcję do wykonania ćwiczenia z zagadnienia zastosowania metod gradientowych do rozwiązania zadania odwrotnego, na przykładzie transferu ciepła w stacjonarnym przypadku dla materiału jednorodnego, do przedmiotu Rozwiązywanie Zadań Odwrotnych. Bazując na niniejszym opisie należy zaprojektować i zaimplementować algorytmy gradientowe w oparciu o kryteria przedstawione w dalszej części tego dokumentu.

1 Metody gradientowe

Wprowadzenie

Wraz z tym dokumentem otrzymali Państwo również załączniki: opis metod:

- BFGS
- L-BFGS-B
- Newton-CG
- gradient prosty
- TNC
- CG

Należy dokładnie zapoznać się z tymi dokumentami, gdyż stanowią one bazę matematyczną, konieczną do rozwiązania postawionego przed Państwem zadania.

Cel ćwiczenia

Celem ćwiczenia jest **zaimplementowanie czterech algorytmów, które korzystając z różnych metod gradientowych rozwiążą problem odwrotny polegający na określeniu warunków brzegowych w jednowymiarowym, jednorodnym problemie transferu ciepła w języku Python**. Efektem końcowym są skrypty, które pozwolą na analizę rozkładu temperatury wzdłuż niejednorodnego pręta oraz w niejednorodnym materiale 2D.

Środowisko pracy

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

Biblioteki

- numpy
- matplotlib
- spicpy
- time

*Katedra Informatyki, Wydział Informatyki i Sztucznej Inteligencji (kgruszka@icis.pcz.pl)

Wymagania funkcjonalne algorytmów

1. Proszę użyć 4 algorytmów w podanej kolejności: BFGS, L-BFGS-B, Newton-CG, gradient prosty.
2. Definicja ilości węzłów siatki: Algorytm musi pozwalać użytkownikowi na zdefiniowanie ilości węzłów siatki N .
3. Warunki brzegowe: Użytkownik musi mieć możliwość zdefiniowania temperatur na obu końcach 1-wymiarowego pręta.
4. Dane o rozkładzie temperatury przygotować korzystając z wcześniej napisanej funkcji MRS.
5. Wizualizacja wyników: temperatury w poszczególnych węzłach powinny być wizualizowane na wykresie, na którym oś X reprezentuje numer węzła, a oś Y temperaturę (matplotlib).
6. skrypt ma wyświetlić otrzymane na drodze optymalizacji warunki brzegowe

Wydajność obliczeniowa

Proszę rejestrować czasy działania wszystkich algorytmów (tylko części z gradientem) i zebrać je w formie wykresu (np słupkowy) na samym końcu raportu. **Testowanie i dokumentacja:**

- Przetestuj algorytm dla różnych wartości ' N ', ' T_0 ', ' T_N ' i ' $tolerance$ '.
- Dokumentujcie każdy test oraz wyniki.

2 Forma i ocena wykonania ćwiczenia

Forma oddania zadania

Zadanie należy oddać w formie pliku źródłowego (.py lub .ipynb) zawierającego implementację algorytmów oraz wygenerowane wykresy wraz z opisami

Dla każdego z algorytmów należy przeprowadzić test dla następujących parametrów symulacji transferu ciepła (1D):

- Liczba węzłów N : 50
- $k_{al} = 237.0$ [W/m·K] (aluminium)
- $k_w = 0.12$ [W/m·K] (drewno)
- $T_0 = 150$
- $T_N = 50$
- $tolerancja = 0.01K$
- $max_iter = 10000$

Ponadto w raporcie (np .ipynb, .pdf) należy umieścić następujące informacje:

- Udokumentowanie przeprowadzonego testu dla parametrów podanych powyżej, wyniki oraz odpowiednie komentarze i wyjaśnienia.
- Uzyskane czasy trwania obliczeń dla części gradientowej
- Wykres przedstawiający rozkład temperatur w pręcie.

Zadanie dodatkowe (nieobowiązkowe) korzystając z jednego z wymienionych algorytmów przeprowadź procedurę odwrotną dla materiału jednorodnego 2D. Za rozwiązanie tego zadania można zdobyć dodatkowe max 5 pkt.

Ocena ćwiczenia będzie bazować na:

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