

Laboratorium z Metod Numerycznych

Zadanie projektowe - wymagania

W implementacji projektów **zakazane** jest używanie zmiennych **symbolicznych** i korzystanie z **obliczeń symbolicznych**.

Materiały do **oceny** z kolejnych etapów projektów należy przekazać osobie prowadzącej zajęcia przed danymi zajęciami rozliczeniowymi. Dokładne terminy oraz sposób przekazania rozwiązań określa osoba prowadząca zajęcia w danej grupie laboratoryjnej.

Raport oraz **Prezentacja** powinny zawierać:

- **dane studenta** (imię, nazwisko, grupa),
- **temat i treść** zadania.

Ocena

Część I:

18 pkt - KOD + skrypt testujący

7 pkt – poprawność metody

3 pkt – estetyka kodu, podział na funkcje (oraz komentarze funkcji), wektoryzacja kodu

8 pkt – **skrypt testujący** generujący przykłady – bardziej zaawansowany, w tym:

- porównanie wyników do innej metody wbudowanej Matlaba/analitycznej;
- testy dla minimum 4 do 6 przykładów (różne, np. mały/duży wymiar macierzy, funkcje o odmiennym charakterze, itp.) pokazujące poprawność zaimplementowanej metody (również taki przykład, dla którego algorytm nie zwraca wyniku – z uzasadnieniem);
- dla podanych przykładów porównać różne rodzaje błędów, zbieżność algorytmu, itp.
- dla podanych przykładów tablicowanie wyników, wizualizacja wyników (grafika);

10 pkt - Raport

5 pkt – opis matematyczny; uzasadnienie i wyprowadzenie wzorów dla algorytmu (*jak opis dla osoby, która nie zna algorytmu*); opis szczególnych przypadków kiedy algorytm nie działa;

2 pkt – opis implementacji, opis najważniejszych funkcji w kodzie, najważniejszych elementów skryptu;

2 pkt – opis wyników uzyskanych dla podanych przykładów i wnioski (2-3 zdania);

1 pkt – styl/poprawność językowa;

Część II:

18 pkt – Prezentacja

4 pkt – zwięzły wstęp – do czego służy algorytm + o twórcy (trochę historii) + cel projektu;

4 pkt – prezentacja metody – na czym polega, krótki opis/omówienie, kluczowe wzory;

4 pkt – kluczowe elementy implementacji metody, najważniejsze funkcje;

6 pkt – prezentacja i analiza wyników (wykresy, animacje?, ciekawe przykłady, ciekawostki, itp.); wykonaj różne serie testów porównujących, np. różne rodzaje błędów, liczbę iteracji, zbieżność algorytmu, istnienie rozwiązania, zależność czasu wykonania obliczeń od rozmiaru macierzy wejściowej, itp.