Lab5C (5 pkt) 8.12.2023

Termin odesłania **15.12.2023 (pt) do godz. 14.15** na platformie **Ms Teams** (we właściwym zespole **lab** przypisanym dla przedmiotu **Programowanie Matematyczne**). **Opóźnione** przesłanie rozwiązania zadania będzie rozliczane zgodnie z regulaminem przedmiotu.

Rozwiązanie zadania tj. wszystkie źródłowe **m-pliki, raport** (**obowiązkowy, w formacie PDF z omówieniem wyników**) i w raporcie **oświadczenie o samodzielności** – całość w formacie **zip** o nazwie `pm5c_swojenazwisko_swojeimie.zip`

Raport (plik **pdf**) powinno być w formacie **A4** i powinno obejmować:
Dane studenta (imię, nazwisko, grupa, data)
Treść zadania (postać rozwiązywanego problemu)
Opis kroków przekształcania zadania, krótki opis algorytmu
Ciekawe przykłady obliczeniowe (również dodatkowo wskazane w treści zadania)
Analizę (omówienie) wyników obliczeniowych, testów
Ponadto (do pliku zip) należy załączyć:
Kody źródłowe wszystkich funkcji/procedur i skryptów (**brak** kompletu jest traktowany jak **brak** przesłania zadania, podobnie kod który **nie działa** bo nie jest kompletny… nie będą przyznane żadne punkty)

Napisz **skrypt**, w którym proszę wykonać całe zadanie **kolejnymi etapami**, wywołać przygotowane funkcje oraz przeprowadzić proponowane testy.

- Wygeneruj **nieosobliwy** kwadratowy układ równań liniowych $Ax = b,\ A \in R^{nxn},\ b \in R^n$, gdzie $A = A^T$ dodatnio określona, $n = 10:10:200$ (*może więcej? Ile?*)

  Wygeneruj symetryczną dod. określoną macierz A z **narzuconymi wartościami własnymi, np**: podaj wektor wartości własnych (całkowitych), utwórz macierz diagonalną D z podanymi wartościami, wylosuj nieosobliwą macierz *V*, *V=orth(V)*, $A = VDV^T$. **W raporcie należy opisać konstrukcję macierzy *A*.**

- Napisz plik `fun.m` definiujący **wypukłą funkcję kwadratową**, której minimalizacja odpowiada **rozwiązywaniu** powyższego układu.
  Oprócz **wartości** funkcji, należy również zwrócić jej analityczny **gradient** oraz **hesjan**.

- Rozwiąż układ wykorzystując `operator \` (rozwiązanie ***xE***)

- Przyjmij $x_0 = [\,0, 0, 0, \dots, 0]^T$
  Zastosuj funkcję `fminunc` do znalezienia **wartości *min*** funkcji oraz **punktu optymalnego** startując z x_0.
  W `optimoptions` ustaw pola:
  `Algorithm: quasi-Newton, Display: iter, GradObj: on`

  Porównaj uzyskane **rozwiązanie *xFminunc*** z dokładnym rozwiązaniem układu.
  Oblicz **normę** różnicy: ***norm(xE-xFminunc)***

- napisać funkcję wykorzystującą **algorytm gradientu sprzężonego FR**
  `[xFR,fval,it]=FR(fun,x0,e)`
  `xFR` RO zadania
  `fval` optymalna wartość funkcji
  `it` liczba iteracji

  Oblicz **krok minimalizacji kierunkowej wg** *(porównaj skuteczność w testach)***:**
  ✓ **analitycznego wzoru** dla funkcji kwadratowej
  ✓ algorytm **Armijo** z **kontrakcją**

  Zbadaj **zależność liczby iteracji** dla rosnącego **parametru *n*** (liczby zmiennych), zależność od **liczby unikalnych wartości własnych**.
  Wykonaj wykresy prezentujące **wartość normy gradientu w kolejnych iteracjach**.
  Oblicz **normę** różnicy: ***norm(xE-xFR)***
  Wnioski

- napisać funkcję wykorzystującą **algorytm najszybszego spadku** **NS**

  `[xNS,fval,it]=NS_eigs(fun,x0,e)`

  `xNS` RO zadania

  `fval` optymalna wartość funkcji

  `it` liczba iteracji

  Niech **`lambda`** – wektor **unikalnych wartości własnych** (uporządkowany rosnąco? Malejąco? Może inny porządek czy ma znaczenie?)
  Do **minimalizacji kierunkowej** należy zastosować kolejne **wartości własne** : $\alpha_k = \lambda_k^{-1}$

  Zbadaj **zależność liczby iteracji** dla rosnącego **parametru *n*** (liczby zmiennych), zależność od **liczby unikalnych wartości własnych**.
  Wykonaj **wykresy** prezentujące **wartość normy gradientu w kolejnych iteracjach**.
  Oblicz **normę** różnicy: ***norm(xE-xNS)***
  Wnioski

- **Opis testów**
- **Wnioski**