Lab5C (6 pkt=1+5) 10.12.2021

Termin odesłania **17.12.2021 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. **Wyjątkowo** termin drugi przypada na **10.01.2022 godz. 14.15**.

Rozwiązanie zadania tj. wszystkie źródłowe **m-pliki, raport** (*obowiązkowy, zawierający oświadczenie o samodzielności*) w formacie **zip** o nazwie `pm5c_swojeimie_swojenazwisko.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 należy załączyć:
Kody źródłowe wszystkich funkcji/procedur i skryptów (**brak** kompletu jest traktowany jak **brak** przesłania zadania w terminie)

Napisz **skrypt**, w którym proszę wykonać całe zadanie i wywołać odpowiednie funkcje.

1 pkt

- Wygeneruj **nieosobliwy** kwadratowy układ równań liniowych $Ax = b$, $A \in R^{nxn}$, $b \in R^n$, gdzie $A = A^T$ dodatnio określona, o losowych **współczynnikach całkowitych** z pewnego przedziału $[p1, p2]$, $n = 10{:}10{:}200$ (*może więcej? Ile?*)

- Napisz plik `fun.m` definiujący **wypukłą funkcję kwadratową** (funkcja posiada argument $\boldsymbol{x}$), której minimalizacja odpowiada **rozwiązywaniu** powyższego układu.
  Oprócz **wartości** funkcji, należy również zwrócić jej analityczny **gradient** oraz **hesjan**.

  Zastosuj f. **`fminbnd`** *(w opcjach ustaw wyświetlanie kolejnych iteracji)*, do znalezienia minimum funkcji dla $d_0 = -\nabla f(x_0)$ w przedziale $[\alpha_0, \alpha_{\max}]$, gdzie lewa granica $\alpha_0 = 0$ odpowiada $x_0 = zeros(n, 1)$, natomiast $\alpha_{\max}$ ustalić na podstawie <u>własnej</u> funkcji `alfa_max.m` z parametrami:
  $\boldsymbol{\alpha_{max}}$**`=alfa_max(@(alfa)fun(x0+alfa*d0),…)`**

W czasie zajęć można przyjąć: α_{max} stałą wartość.

  Narysować wykres (funkcja `fplot`) w przedziale $[\alpha_0, \alpha_{\max}]$

  <u>wskazówka:</u>
  `x0=ustalone`
  `d0=ustalone`
  `fplot(@(alfa)fun(x0+alfa*d0),`$[\alpha_0, \alpha_{max}]$**`)`**

Przyjąć dokładność obliczeń `e=1e-4`

- napisać funkcję wykorzystującą <u>algorytm **złotego** podziału</u> (zdefiniuj funkcję `gold.m`):

  `[alfa_gold,it]=gold(@(alfa)fun(x0+alfa*d0),` α_0 `,` $\alpha_{\max}$ `,e)`

  Podaj wartość kroku `alfa_gold=?`
  Ile wykonano iteracji `it=?`

- napisać funkcję wykorzystującą <u>algorytm **Armijo** z kontrakcją</u> (zdefiniuj funkcję `Armijo.m`):

```
[alfa_Armijo,it]=Armijo(@(alfa)fun(x0+alfa*d0),…inne potrzebne parametry-
jakie? uzasadnij);
```

  Podaj wartość kroku `alfa_Armijo=?`
  Ile wykonano iteracji `it=?`

Podaj **3 rozwiązania** dla **alfa**: z fminbnd, gold oraz Armijo

5 pkt

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

- 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(xExact-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

  Do min. kierunkowej (przyjmij **`e=1e-4`**) oblicz **krok minimalizacji kierunkowej wg** *(porównaj skuteczność w testach)***:**
  - ✓ **analitycznego wzoru** dla funkcji kwadratowej
  - ✓ algorytm **złotego podziału**
  - ✓ algorytm **Armijo z kontrakcją**

  Przyjmij dokładność badania stacjonarności **`e=1e-6`** (może lepsza dokładność? może jeszcze inne dodatkowe warunki stopu?)

  Wykonaj obliczenia dla funkcji **`fun`** (definiującej wypukłą funkcję kwadratową z pierwszej części zadania).
  Zbadaj **zależność liczby iteracji** dla rosnącego parametru ***n*** (liczby zmiennych).
  Wykonaj **wykresy** prezentujące **wartość normy gradientu w kolejnych iteracjach**.
  Oblicz **normę** różnicy: ***norm(xExact-xFR)***

- Wykonaj **testy, zweryfikuj** warunek stopu:
  - ✓ Dla startowego problemu z macierzą A
    - Jaka jest liczba iteracji algorytmu **FR**?
    - Wykonaj **wykres** prezentujący **wartość normy gradientu w kolejnych iteracjach**.
    - **Wnioski**

  Następnie (możesz zaproponować **inny algorytm** generowania)
  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*.**

  - ✓ Ustaw wszystkie wartości własne na wspólną wartość, np. $\lambda_i = t,\ \ t > 0,\ \ i = 1, \dots, n$
    - Jaka jest liczba iteracji algorytmu **FR**?
    - Wykonaj **wykres** prezentujący **wartość normy gradientu w kolejnych iteracjach**.
    - **Wnioski**

  - ✓ Ustaw wszystkie wartości własne na wspólną wartość, np. $\lambda_i = t,\ t > 0\ \ i = 1, \dots, n-1$
    - $\lambda_n = z,\ \ gdzie\ z\ znacząco\ odbiega\ od\ wartości\ t$
    - Jaka jest liczba iteracji algorytmu **FR**?
    - Wykonaj **wykres** prezentujący **wartość normy gradientu w kolejnych iteracjach**.
    - **Wnioski**

  - ✓ Ustaw ***k*** **klastrów** wartości własnych, np. dla ***k=5*** można przyjąć skokowe wartości 10, 20, 30, 40, 50 lub inne
    - Jaka jest liczba iteracji algorytmu **FR** dla różnych wartości ***k***? Czy można ustalić ***hipotezę***?
    - Wykonaj **wykres** prezentujący **wartość normy gradientu w kolejnych iteracjach**, dla przykładowych wartości k=5÷10.
    - **Wnioski**

- **Opis testów**
- **Wnioski**