Zadanie

2 pkt

1. Napisz plik **fun.m** definiujący funkcję zmiennej ***x***:
   $f(x) = x_1^2 + x_2^2 + 3x_3^4 + 2x_1^2 x_2^2 + x_1 x_2 + 2x_2 x_3 + x_1 + 2x_2 + 3x_3$
   Funkcja zwraca jej ***wartość***, ***gradient*** i ***hesjan***.

   Zastosuj f. **`fminbnd`** (ile iteracji?), do znalezienia ***min*** funkcji w kierunku $d_0 = -\nabla f(x_0)$, dla $x_0 = [\,3;\ -1;\ -1\,]$ w przedziale $[\alpha_0, \alpha_{max}]$, gdzie lewa granica $\alpha_0 = 0$ odpowiada x_0, natomiast α_{max} ustalić na podstawie swojej własnej funkcji **alfa_max.m** z parametrami:

   ```
   [a1,a2,a3]=alfa_max(@(alfa)fun(x0+alfa*d0), …inne parametry…);
   ```
   α_{max} =a3

   Narysować wykres (f. **fplot**) w przedziale $[\alpha_0, \alpha_{max}]$

   wskazówka:
   ```
   x0=ustalone
   d0=ustalone
   fplot(@(alfa)fun(x0+alfa*d0), [α0, αmax]);
   ```

Przyjąć dokładność obliczeń `e=1e-4`

1,5 pkt

2. napisać funkcję wykorzystującą aproksymację paraboliczną (zdefiniuj funkcję **parabola.m**):

   ```
   [alfa1,n_iter]=parabola(@(alfa)fun(x0+alfa*d0),p,e);
   parametry:
   -funkcja
   ```
   – $p = [a1, a2, a3]$ `odpowiada 3 punktom kontrolnym paraboli`
   ```
   -dokładność obliczeń: e
   ```

   Podaj wartość kroku alfa=? Ile wykonano iteracji ?

1,5 pkt

3. napisać funkcję wykorzystującą algorytm Newtona (zdefiniuj funkcję **Newton.m**):
   ```
   alfaN=Newton(@(alfa)fun(x0+alfa*d0), α0,d0,e)
   ```

   Podaj wartość kroku `alfaN` =?
   Ile wykonano iteracji ? Wyświetl kolejne przybliżenia.