7 Instrukcja do ćwiczenia nr 7: Aproksymacja nieliniowa - kontynuacja.

7.1 Wstęp

1. **Wnioski z poprzedniego ćwiczenia**
   – Cechy aproksymacji liniowej:

   - ograniczony wybór klasy funkcji ograniczony do funkcji liniowo zależnych od parametrów,
   - ograniczony wybór normy - do średniokwadratowej,
   - prosty algorytm, jednoznaczne rozwiązanie - istnieje dokładnie jedno (jeżeli tylko funkcje $f_i(x)$, które tworzą bazę, są liniowo niezależne).
   - problem wyboru liczby parametrów - ograniczenia: dalsze - błąd numeryczny, bliższe - nadmierne dopasowanie (do zakłóceń).

2. Ostatnia cecha pozostanie aktualna także w aproksymacji nieliniowej.

3. Czego oczekujemy po aproksymacji nieliniowej?
   Braku ograniczeń w wyborze klasy funkcji i normy odległości, a tym samym - większej możliwości odwzorowania natury zjawiska.

4. Czym prawdopodobnie zapłacimy?

   - złożonością algorytmu wyznaczania optymalnych wartości parametrów, a zatem czasem obliczeń,
   - niepewnością optymalności rozwiązania.

5. Cele ćwiczenia – poznanie trudności pojawiających się w problemach optymalizacji bez ograniczeń:

   - minima lokalne,
   - zależność rozwiązania od punktu startowego,
   - dobór opcji dla kryterium stopu algorytmu optymalizacji,
   - wybór algorytmu optymalizacji,
   - w aproksymacji nieliniowej – dodatkowo wybór funkcji aproksymującej i normy odległości.

7.2 Plan

1. Ilustracja narzędzi MATLABa (funkcji dla optymalizacj statycznej: `fminsearch, lsqnonlin, fminunc`) na przykładzie aproksymacji przebiegu fali wezbraniowej w rzece.
   Problem wyboru klasy funkcji aproksymującej i liczby parametrów:
   - porównanie 3 wersji funkcji aproksymującej,
   - przykład funkcji wymiernej.
2. Aproksymacja fazy narastania epidemii
   - prosta funkcja aproksymująca dla początkowej fazy epidemii,
   - uelastycznienie (przez zwiększenie liczby parametrów) funkcji aproksymującej dane w całym zakresie danych,
   - obserwacja problemów optymalizacji w przestrzeni wielu zmiennych (wpływu wyboru punktu startowego, kryterium stopu itp.),
   - porównanie wyników otrzymanych dla różnych norm odległości.
3. Budowa modelu epidemicznego (z uwzględnieniem stopniowego nabywania odporności w społeczeństwie) z możliwością ekstrapolacji symulacji poza zakres danych pomiarowych – przy użyciu SIMULINKa – na następnym ćwiczeniu.

7.3 Przebieg ćwiczenia

Uwaga: W plikach `aprox_pocz.m` oraz `aprox_total.m` proszę ustawić zmienną `wersja` na 2019 (chyba, że ktoś uzywa MATLABa w wersji z np. 2013 roku).

1. Fala wezbraniowa na rzece.
   Dane są takie same jak w ćwiczeniu 6 – są to pomiary przepływu w rzece Rabie w okresie wiekszego opadu deszczu i późniejszego wezbrania Raby. Przepływ q był mierzony (a włściwie obliczany na postawie poziomu wody) co godzinę. Dla naszych celów aproksymacji wykorzystamy część tych danych, odpowiadających fazie opadania poziomu wody, tj. począwszy od 19 pomiaru (stąd oznaczenia w skrypcie : t19, q19 itp.)

   a. Porównanie funkcji aproksymujących tej samej klasy, ale z innymi parametrami:
   Wykonując skrypt `fala_wezbraniowa.m` jest możliwość wyboru jednej z 3 funkcji wykładniczych (ewentualnie z dodaniem stałej):

$$f_1(t) = a_1 + \exp(a_2 t),$$

$$f_2(t) = a_1 \exp(a_2 t),$$

$$f_3(t) = a_1 + a_2 \exp(a_3 t),$$

Na razie pomijamy funkcję wymierną $f_4(t)$.
W każdym przypadku, w wektorze `a0` jest zadawany punkt startowy wartości parametrów (mam nadzieję, że zmiana nie będzie trudna).

Wykonując skrypt jest też możliwość wyboru algorytmu opymalizacji – szukania takich wartości parametrów funkcji, dla których odległość funkcji od danych jest minimalna. W każdym przypadku zapisanym w skrypcie odległośc ta jest mierzona sumą kwadratów odległości.
Do wyboru są 3 algorytmy:

- `fminsearch` – funkcja stosująca bezgradientowy algorytm Neldera-Meada - opis np. na wykładzie,
- `lsqnonlin` – funkcja dla nieliniowych problemów najmniejszych kwadratów,
- `fminunc` – funkcja stosująca różne algorytmy gradientowe (gradient obliczany numerycznie lub analitycznie) przeznaczona dla ogólnych zadań optymalizacji nieliniowej bez ograniczeń.

Dokładny opis tych funkcji jest w pomocniku MATLABa dostępnym także w internecie (hasło np: matlab lsqnonlin).

b. Przykład funkcji wymiernej.
Czwartą opcją wyboru funkcji jest funkcja wymierna. Pojawia się problem wyboru stopnia wielomianu w liczniku i mianowniku. Czym należy się kierować?

- Obydwa nie powinny być wysokie ze względu na niebezpieczeństwo nadmiernego dopasowania,
- O tym, który ma być wyższy powinno decydować zachowanie się funkcji dla dużych t – co umożliwiłoby ekstrapolację poza zakres danych.

W skrypcie jest zapisany jeden przykład

$$f_4(t) = \frac{a_1 t + a_2}{a_3 t^2 + a_4 t + a_5},$$

ale można go łatwo zmodyfikować (pamiętając tylko o dostosowaniu rozmiaru wektora wartości początkowych `a0` do liczby parametrów funkcji).
Należy ocenić jakość aproksymacji i ewntualnie wybrać inną postać.

c. Wskazana byłaby samodzielna próba aproksymacji całego zakresu danych (od pierwszego do ostatniego pomiaru).

2. Epidemia

a. Naszym zadaniem jest aproksymacja przebiegu liczby zdiagnozowanych dodatnio zakażonych wirusem covid-19 w Polsce. Oficjalne dane są w skrypcie `aprox_pocz.m`. Proponuję skopiować z niego wektor LZ i narysować wykres plot(LZ,'o'),

Czy można wyodrębnić dwa zakresy danych, w których widoczne są rózne rodzaje zależności – nieliniową i w przybliżeniu liniową?

Nasuwają się dwa sposoby podejścia do rozwiązania problemu aproksymacji tych danych:

- **Zaproponować jedną funkcję dla aproksymacji całego przedziału danych.** Jakie trudności się z tym wiążą. Aby znaleźć aproksymację bardziej złożonej zależności z małym błędem, trzeba przewidzieć funkcję z większą liczbą parametrów. Z pierwszej części ćwiczenia spodziewam się wniosku, że zbieżność algorytmu szukania optymalnych parametrów zależy m.in. od wyboru punktu startowego (jeżeli nie było to widoczne, to będzie można się o tym przekonać w ostatniej części tego ćwiczenia). Trudność znalezienia dobrego punktu startowego wzrasta wraz z wzrostem liczby parametrów, co prowadzi do poszukiwania rozwiązania problemu optymalizacji w przestrzeni o tylu wymiarach ile jest parametrów.
  Przewidując te problemy proponuję inne podejście:
- **Zaproponować prostą funkcję dla aproksymacji tylko początkowego odcinka nieliniowego.** Wtedy łatwo znaleźć punkt startowy dla poszukiwań optymalnych parametów tej części aproksymacji, a w następnym etapie „uelastycznić” funkcję aproksymującą dodając kolejne parametry. Przy wyborze punktu startowego rozszerzonej funkcji będziemy już dysponować wiedzą o przybliżonych wartościach „starych” parametrów funkcji – co ułatwi trafienie w dobry punkt startu.

Teraz dygresja:
Zjawiska rozwoju epidemii opisuje się zwykle równaniami różniczkowymi, wg których przyrost liczby zakażonych zależy proporcjonalnie od liczby zakażonych.

$$\frac{dZ}{dt} = \beta Z,$$

gdzie Z jest liczbą zarażonych, a β współczynnikiem proporcjonalności.
Jest to liniowe równanie różniczkowe, a jego rozwiązanie jest funkcją wykładniczą $Z(t) = C \exp(\beta t)$.
Koniec dygresji.
Dlatego proponuję aproksymować pierwszą część zależności funkcją podobnej postaci, jak w przypadku fali wezbraniowej.

W skrypcie `aprox_pocz.m` jest możliwość wyboru 3 funkcji i 3 algorytmów – analogicznie jak w pierwszej części ćwiczenia. Należy sprawdzić, czy rzeczywiście funkcja wykładnicza może dobrze aproksymować dane.

b. Może zrodzić się pomysł, żeby do aproksymacji danych w całym zakresie użyć funkcji sklejanej z części wykładniczej i liniowej. Jakie są wady takiego podejścia:

- nie odzwierciedla rzeczywistej zależności - trudno byłoby uzasadnić dlaczego liczba zarażonych ma rosnąć liniowo,
- ponad wszelką wątpliwość takie rozwiązanie nie dawałoby szansy na prawidłową ekstrapolację (poza zakres danych).

Dlatego w dalszej części proponuję aproksymację cały czas jedną funkcją, ale o zmieniającej się w czasie wartości współczynnika β.

Druga dygresja:
Od czego zależy wartość β. W pierwszym przybliżeniu od liczby ludzi, z którymi zakażony się zetknął. Zatem można oczekiwać, że liczba ta (a za nią i współczynnik β) zmieni się po wprowadzeniu ograniczeń w kontaktowaniu się obywateli (14 marca – zamknięcie szkół, 26.marca – tzw. lockdown).
Koniec drugiej dygresji.
Wg mojego założenia, współczynnik β powinien stopniowo zmieniać się od wartości początkowej (bliskiej tej odczytanej z aproksymacji pierwszej fazy) do wartości na niższym poziomie. Jaka funkcja zmienia się od jednego poziomu do innego poziomu? Zaproponowałem funkcję $\arctan(t)$ odpowiednio przeskalowany i przesunięty. Wtedy funkcja aproksymująca cały zakres danych mogłaby mieć postać np:

$$f(t) = a_1 + a_2 \exp\left[(a_3 \arctan(a_4 t + a_5) - a_6)\, t\right]$$

Czy jest to słuszny i jedyny wybór? NA PEWNO NIE! – zachęcam prób z innymi propozycjami.

Mamy przypadek funkcji z większą liczbą parametrów.

Skrypt `aprox_total.m` ma zapisaną jedną funkcję aproksymującą i jeden punkt startowy (początkowe wartości parametrów tej funkcji). W czasie wykonywania jest mozliwość wyboru algorytmu – jak w poperzedniej części ćwiczenia.

Co należy zrobić?

i. Zauważyć, jak ważny jest dobry wybór punktu startowego algorytmu optymalizacji.
ii. Zwrócić uwagę na przyczynę stopu: Jeżeli przyczyną jest wykonanie maksymalnej liczby iteracji lub liczby obliczeń funkcji celu (obliczającej odległość aproksymacji od danych), to należy zwiększyć przyjęte w opcjach limity. W przypadku podejrzenia dojścia do minimum lokalnego należy zmienić punkt startowy.
iii. Włączyć opcję `Display` funkcji optymalizacji - patrz pomocnik MATLABa. Jeżeli postęp w szukaniu punktu optymalnego jest mały, to z osiągniętego punktu wystartować z użyciem innego algorytmu. Często dobre efekty daje taka strategia: kilkaset kroków algorytmem gradientowym i w przypadku zwolnienia postępu – kontynuacja algorytmem bezgradientowym i ponownie, w przypadku zwolnienia nowa zamiana algorytmu itd.
iv. Przeprowadzić aproksymację z inną normą odległości.
v. Zaproponować inną funkcję aproksymującą o niewielu parametrach

c. Należy rozróżnić dwa zadania (na przyszłość):

- Identyfikacja krzywej ilustrującej pewną zależność– w naszym przypadku: Jak w czasie zmienia się liczba osób zakażonych W tym przypadku chodzi o znalezienie funkcji, która dobrze przybliża przebieg zjawiska (cel ćwiczenia nr 7).

- Budowanie modelu epidemicznego. Zbudowanie modelu to coś więcej: opis matematyczny, który odzwierciedla rzeczywisty proces, a aproksymacja służy tylko kalibracji parametrów. Budowa modelu ma szersze zastosowanie - równania powinny być uniwersalne (dla każdej pandemii), a parametry wyliczane dla konkretnego przypadku
  I to jest zarys zadania na kolejne ćwiczenia – w SIMULINKu.

7.4 Skrypty

1. `fala_wezbraniowa.m`
2. `aprox_pocz.m`
3. `aprox_total.m`

7.5 Wytyczne do sprawozdania

1. Fala wezbraniowa

   a. Wybór funkcji aproksymującej i liczby parametrów:

      i. Wnioski odnośnie do funkcji wykładniczej: czy wystarczą dwa parametry, jaka minimalna liczba parametrów jest wystarczająca, ew. propozycje innych funkcji z małą liczbą parametrów.

      ii. Funkcja wymierna - Czy zaproponowana w skrypcie postać z 5 parametrami jest poprawna? tzn. jeżeli algorytm znajdzie optymalny zestaw 5 parametrów, to czy to będzie jedyny optymalny, czy będzie ich nieskończenie wiele? Jeżeli prawdziwa jest druga odpowiedź, to jak doprowadzić do jednoznaczności rozwiązania optymalnego?

      iii. Ew. wyniki aproksymacji wielomianowej,

      iv. Ew. wyniki aproksymacji całej fali wezbraniowej (nie tylko fazy opadania).

   b. Porównanie algorytmów dla wybranej funkcji aproksymującej (informacje o zastosowanym algorytmie są dostępne w polu `algorithm` i `iterations` w strukturze output w parametrach wyjsciowych funkcji szukającej minimum (`fminsearch,lsqnonlin, fmincon`).
      Dla dociekliwych: w funkcji np. `lsqnonlin` można użyć opcji wymuszenia wyboru algorytmu.
      Kryterium porównania może być czas obliczeń.
      Uwaga: W skrypcie czas obliczeń oblczeń jest mierzony parą „funkcji" `tic, toc`. Wynik może być inny, gdy mierzymy czas pierwszego wykonania skryptu po jego modyfikacji i kolejnego wykonania (w kolejnych wykonaniach nie ma etapu tłumaczenia kodu i czas jest krótszy).

Porównania, o których tu mowa można przeprowadzić albo dla danych o fali wezbraniowej albo dla danych o epidemii (uzywając skryptu `aprox_total.m`) – wtedy należy pominąć pkt 2 a.).

2. Epidemia

   a. Porównania algorytmów (czas obliczeń, ew. odległość punktu startowego od rozwiązania, która nie powoduje utraty zbieżności).

   b. Porównanie wyników otrymanych przy stosowaniu róznych norm.
   Które z 3 algorytmów stosowanych w załączonych skryptach pozwalają na zastosowanie innej normy niż średniokwadratowa?

   c. Inne propozycje funkcji aproksymującej – ewentualnie porównanie wyników (błędu aproksymacji i liczby parametrów).

W przypadku niejasności (a może być ich wiele) proszę je zgłaszać mailowo pamiętając o dopisku w temacie wiadomości: PYTANIE.