Sprawozdanie z laboratorium nr 7
Aproksymacja nieliniowa - kontynuacja.

Emilia Mączka, Marcin Sawczuk, Daniel Warloch, Weronika Wisz

Spis treści

1 Fala wezbraniowa

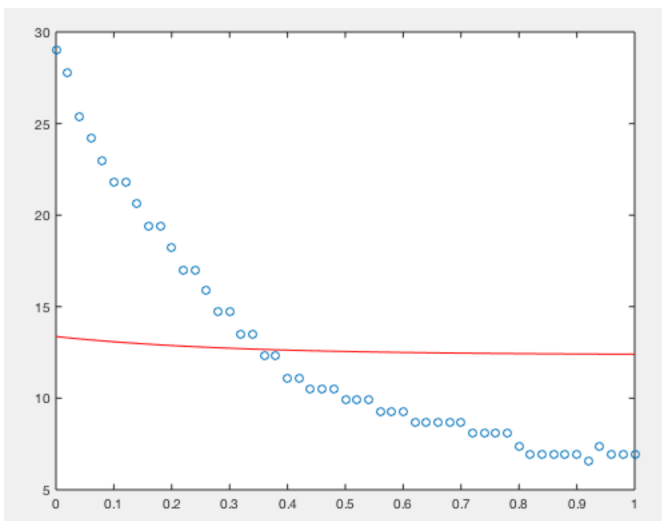
1.1 Wybór funkcji aproksymującej i liczby parametrów:

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

Przeanalizujemy po kolei trzy funkcje wykładnicze, dwie z dwoma parametrami i jedną z trzema. Do optymalicacji parametrów stosujemy funkcje fminsearch, lsqnonlin i fminunc.

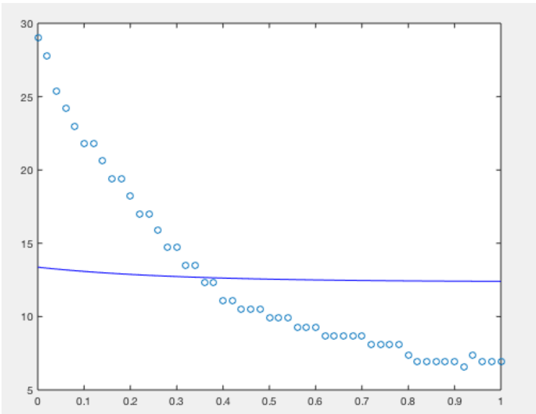
- $f_1(x) = a_1 + exp(a_2 t)$

  Wynik dla fminsearch



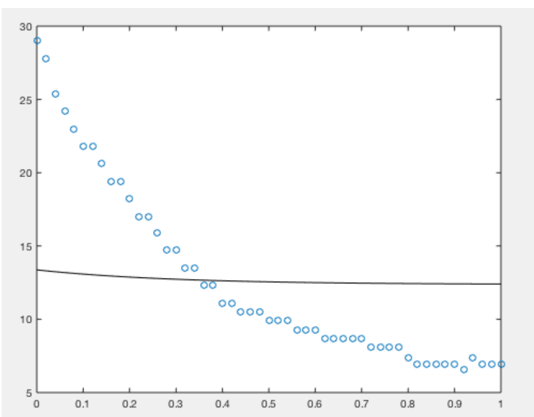

```
parametry2 =
   1.2370e+01  -3.3868e+00
ans =
    'Optimization terminated:
      the current x satisfies the termination criteria using OPTIONS.TolX of 1.000000e-04
      and F(X) satisfies the convergence criteria using OPTIONS.TolFun of 1.000000e-04
     '
```

Wynik dla lsqonlin




```
Local minimum found.

Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.

<stopping criteria details>

parametry1 =

   12.3697   -3.3868


ans =

    '
     Local minimum found.

     Optimization completed because the size of the gradient is less than
     the value of the optimality tolerance.

     <stopping criteria details>

     Optimization completed: The first-order optimality measure, 8.437141e-07,
     is less than options.OptimalityTolerance = 1.000000e-06.

     '
```

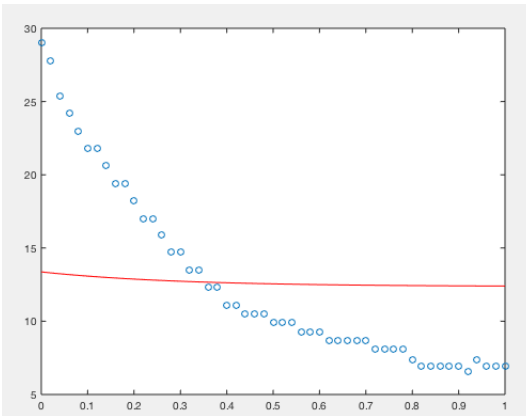
Wynik dla fminunc




```
Local minimum found.

Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.

<stopping criteria details>

a =

   12.3697   -3.3868


fval =

   1.7377e+03


exitflag =

     1


output =

  struct with fields:

       iterations: 16
        funcCount: 60
         stepsize: 2.5044e-05
     lssteplength: 1
    firstorderopt: 8.3882e-05
        algorithm: 'quasi-newton'
          message: '↵Local minimum found.↵↵Optimization completed because the size of the gradient is less than↵the v
```

```
grad =

    1.0e-04 *

   -0.8388
   -0.3604
```

Wyniki nie są zadowalające, przybliżenie funkcji znacząco odbiega od rzeczywistości. Powodem tego może być źle dobrany punkt startowy, algorytm osiągnął minimum lokalne. Zmienimy punkt startowy i jeszcze raz zaobserwujemy co dzieje się z wykresem przybliżenia. Korzystamy już tylko z funkcji fminsearch i sprawdzamy wyniki dla różnych punktów startowych. Poniżej przykład dla a0 = [10,10].

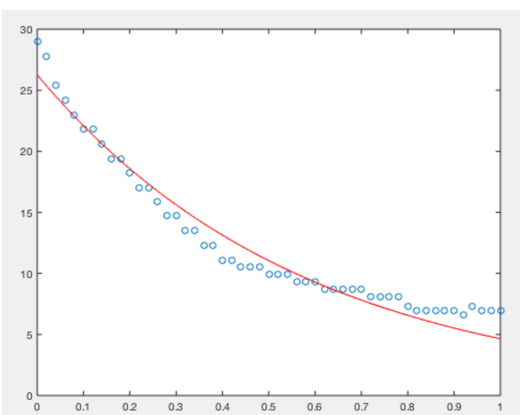



```
parametry2 =
   1.2370e+01  -3.3868e+00
ans =
    'Optimization terminated:
      the current x satisfies the termination criteria using OPTIONS.TolX of 1.000000e-04
      and F(X) satisfies the convergence criteria using OPTIONS.TolFun of 1.000000e-04
     '
```

Jak można zauważyć zmiana punktu startowego nie pomogła, przybliżenie nadal odbiega od rzeczywistości. Możemy wyciągnąć z tego wniosek, że przybliżenia funkcją $f_1(x) = a_1 + exp(a_2 t)$ z dwoma parametrami są niedokładne.

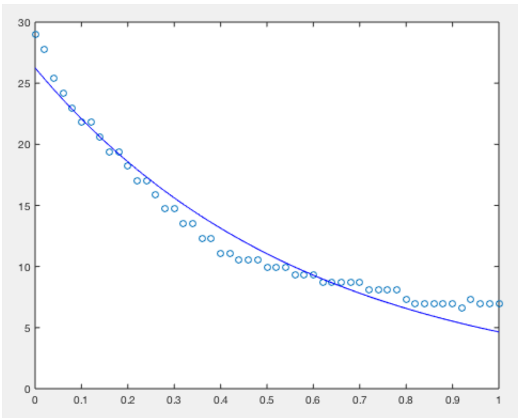
- $f_1(x) = a_1 \cdot exp(a_2 t)$

Wynik dla fminsearch



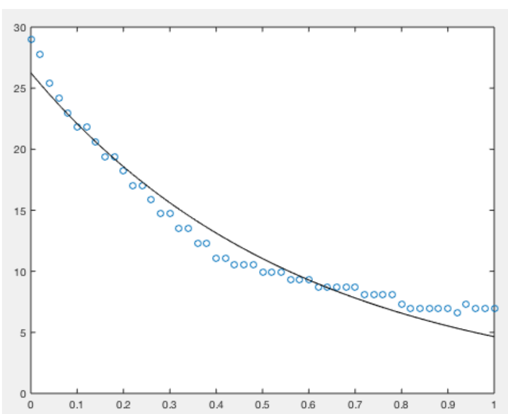

```
parametry2 =

   26.2541   -1.7319


ans =

    'Optimization terminated:
      the current x satisfies the termination criteria using OPTIONS.TolX of 1.000000e-04
      and F(X) satisfies the convergence criteria using OPTIONS.TolFun of 1.000000e-04
     '
```

Wynik dla lsqonlin




```
Local minimum possible.

lsqnonlin stopped because the final change in the sum of squares relative to
its initial value is less than the value of the function tolerance.

<stopping criteria details>

parametry1 =

   26.2541   -1.7319


ans =

    '
     Local minimum possible.

     lsqnonlin stopped because the final change in the sum of squares relative to
     its initial value is less than the value of the function tolerance.

     <stopping criteria details>

     Optimization stopped because the relative sum of squares (r) is changing
     by less than options.FunctionTolerance = 1.000000e-06.

     '
```

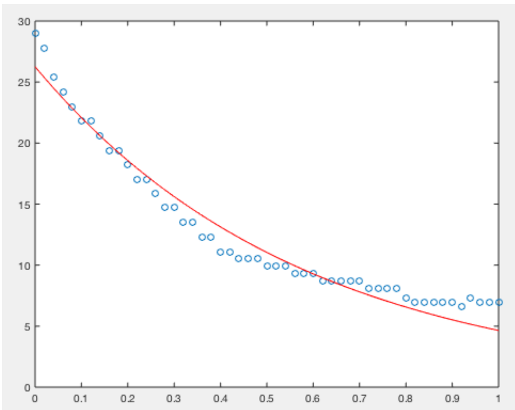
Wynik dla fminunc




```
Local minimum found.

Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.

<stopping criteria details>

a =

   26.2541   -1.7319


fval =

   75.1496


exitflag =

     1


output =

  struct with fields:

       iterations: 15
        funcCount: 54
         stepsize: 1.0560e-04
     lssteplength: 1
    firstorderopt: 0.0015
        algorithm: 'quasi-newton'
          message: '↵Local minimum found.↵↵Optimization completed because the size of the gradient is less than↵the val
```

```
grad =

    0.0003
    0.0015
```

Wykresy przybliżenia są dokładniejsze niż w pierwszym przypadku, nie są jednak idealne. Możemy spróbować zmienić punkt startowy i zaobserwować czy pomoże nam to zwiększyć dokładność. Tak prezetuje się wykres dla a0= [10,10] korzystając z fminsearch.

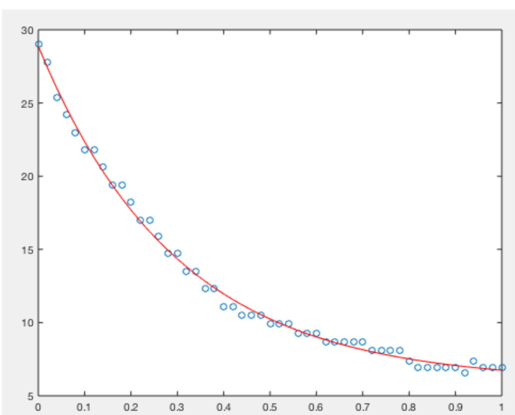



```
parametry2 =

   26.2541   -1.7319


ans =

    'Optimization terminated:
      the current x satisfies the termination criteria using OPTIONS.TolX of 1.000000e-04
      and F(X) satisfies the convergence criteria using OPTIONS.TolFun of 1.000000e-04
     '
```

Dokładność po zmianie punkty startowego nie zwiększyła się. Przybliżenia funkcją $f_1(x) = a_1 \cdot exp(a_2 t)$ z dwoma parametrami również nie zapewnia dobrej dokładności.

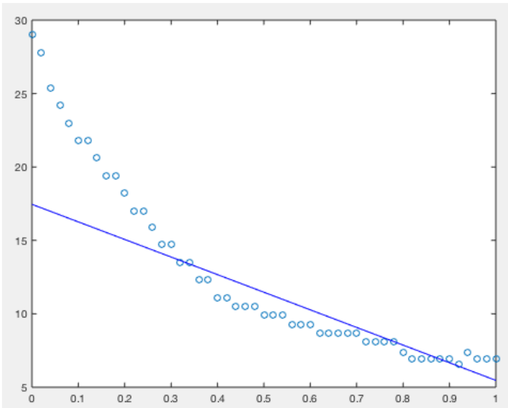
- $f_1(x) = a_1 + a_2 \cdot exp(a_3 t)$

  Wynik dla fminsearch



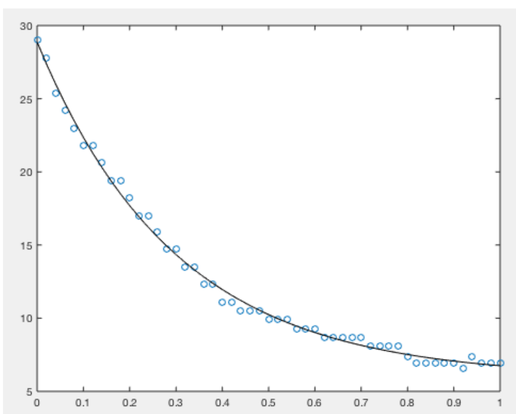

```
parametry2 =

    5.9402   22.9283   -3.3444


ans =

    'Exiting: Maximum number of function evaluations has been exceeded
              - increase MaxFunEvals option.
              Current function value: 7.817416
     '
```

Wynik dla lsqonlin




```
Solver stopped prematurely.

lsqnonlin stopped because it exceeded the function evaluation limit,
options.MaxFunctionEvaluations = 3.000000e+02.


parametry1 =

   1.0e+03 *

    1.9830   -1.9656    0.0000


ans =

    '
     Solver stopped prematurely.

     lsqnonlin stopped because it exceeded the function evaluation limit,
     options.MaxFunctionEvaluations = 3.000000e+02.

     '
```

Wynik dla fminunc




```
Local minimum found.

Optimization completed because the size of the gradient is less than
the value of the optimality tolerance.

<stopping criteria details>

a =

    5.9405   22.9282   -3.3445


fval =

    7.8174


exitflag =

     1


output =

  struct with fields:

       iterations: 27
        funcCount: 120
         stepsize: 0.0015
     lssteplength: 1
    firstorderopt: 6.6192e-05
        algorithm: 'quasi-newton'
```

```
grad =

   1.0e-04 *

    0.6619
   -0.0495
    0.2680
```

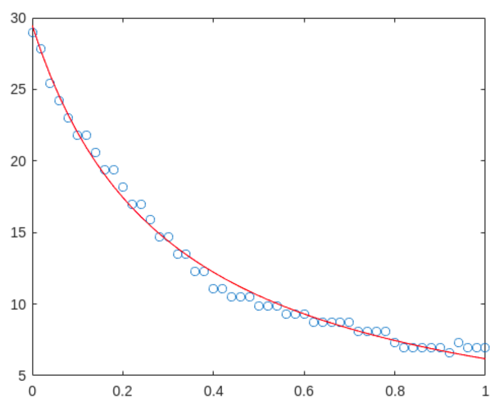
Wynik dla funkcji lsqonlin odbiega mocno od funkcji rzeczywistej, prawdopodobnie dla tej funkcji źle dobrany został punkt startowy, ale korzystając z pozostałych (fminsearch, fminunc) z dużą dokładnością otrzymujemy przybliżenie fukcją $f_1(x) = a_1 + a_2 \cdot exp(a_3 t)$ z trzema parametrami.

Wniosek: Dwa parametry nie wystarczą aby osiągnąć dobre przybliżenie.

1.1.2 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?

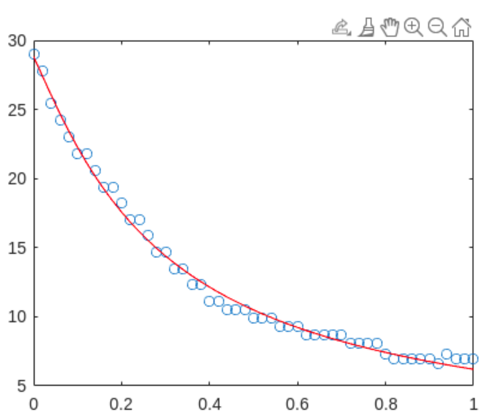
Próbujemy dobrać różne punkty startowe.
Mimo, że zmieniamy je, to można zauważyć, że otrzymano bardzo podobne przybliżenia. Parametry się różnią, jednak nie wpływa to znacząco na aproksymacje, można się spodziewać, że takich optymalnych parametrów będzie nieskończenie wiele.

a0 = [1, 0, 1, 1];



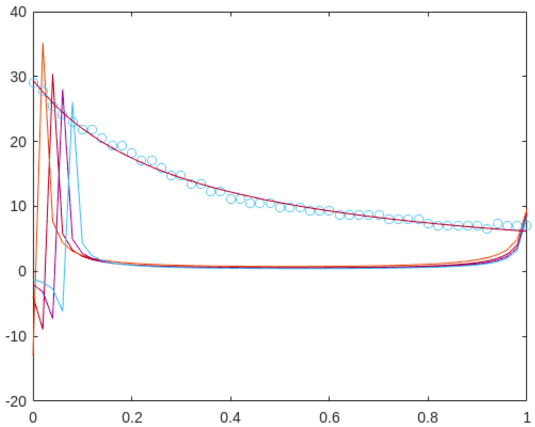

```
parametry2 =

   29.4748    0.0556    3.3689    0.4131
```

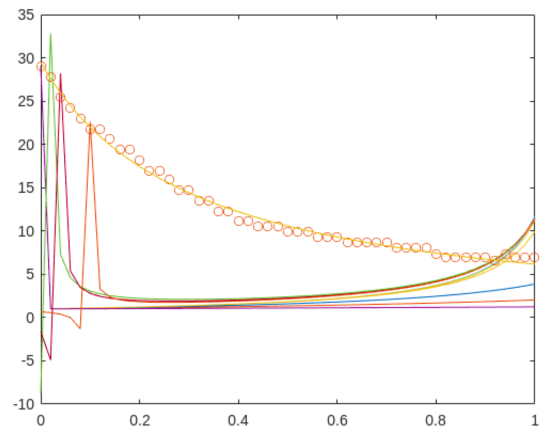
a0 = [100, 120, 150, 100];




```
parametry2 =

   28.8304  183.7238    8.7544   24.5099
```

Zmieniając natomiast liczbę parametrów otrzymujemy nieco inne wyniki. Poniżej znajdują się wykresy które korzystają z trzech (wykres pierwszy) oraz czterech parametrów (wykres drugi).



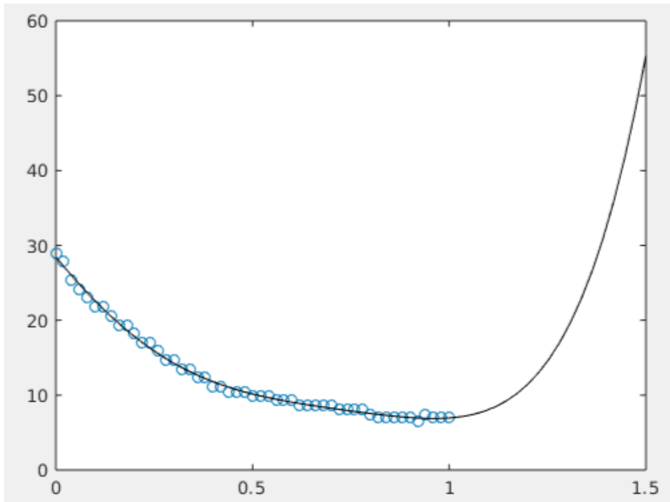



```
parametry2 =

   29.4754    3.3676    0.4054
```

```
parametry2 =

   29.4429    3.4544    0.8777  264.6007
```

Wniosek: Zmniejszanie liczby parametrów prowadzi nas do jednoznaczności rozwiązanie. Musimy dobrać wystarczającą liczbę parametrów, tak aby funkcja dobrze przybliżała, ale także nie za dużą, aby rozwiązanie było jednoznaczne.

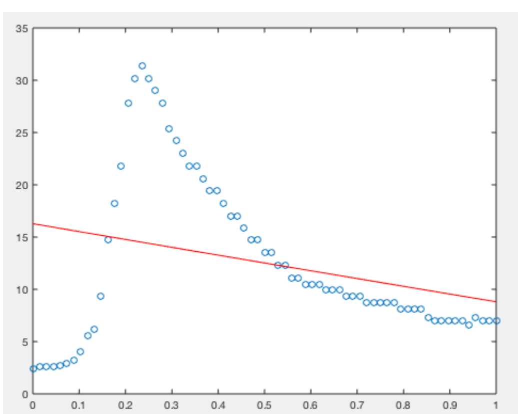
1.1.3 Ew. wyniki aproksymacji wielomianowej,

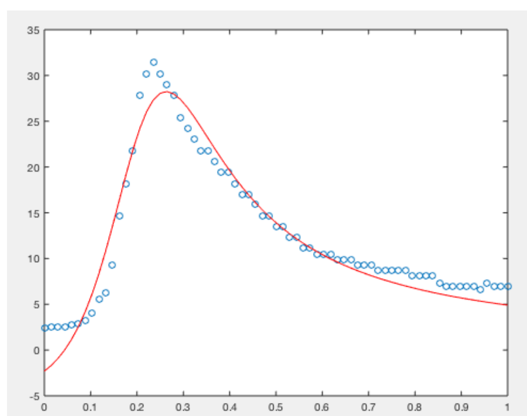


Jeżeli aproksymujemy funkcją wielomianową dla znanych danych to przybliża ona z dobrą dokładnością. Jeżeli jednak chcemy przeprowadzić ekstrapolacje to funkcja wielomianowa się nie sprawdzi co widać na powyższym rysunku.

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

Wyniki aproksymacji liniowej dla funkcji od lewej wykładniczej $f_1(x) = a_1 + a_2 \cdot exp(a_3 t)$ i wymiernej $f(t) = (a_1 * t_n orm + a_2)/(a_3 \cdot t_n orm^2 + a_4 \cdot t_n orm + a_5)$ dla całej fali wezbraniowej.





1.2 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

Modyfikacja algorytmu poprzez dodanie funkcji wskazującej czas wykonywania i liczbę wywołań funkcji pozwoliła w miarę efektywnie porównać algorytmy. Można zauważyć, że metoda fminsearch jest najszybsza mimo dużej ilości wywołań (ponad 1000 w czasie mniej niż 0.01 sekundy). Inaczej prezentują się dwie pozostałe metody. Funckja lsqnonlin mieści się w 0.07 sekundy jednak jest to przy niecałych 200 wywołaniach, natomiast fminunc wykonuje ich więcej - 275, ale za to w dwukrotnie dłuższym czasie.

Fminsearch nie korzysta z pochodnych. Pozostałe dwie oczekują różniczkowalności funkcji optymalizowanej, w zamian za to pozwalają na nałożenie ograniczeń. Jednakże dla prostego problemu jaki badamy, pierwsza funkcja w zupełności wystarcza, wielokrotne wywoływanie oryginalnej funkcji nie jest tu problemem.

Ponieważ sprawdzamy zachowanie dla konkretnego przypadku, znaczenie tu ma czas. Niezależność problemu wymagałaby skupienia sie na kryterium wykorzystującego liczbę wywołań funkcji, by stwierdzić, który algorytm jest lepszym rozwiązaniem.
Reasumując, fminsearch to najlepsza opcja rozwiązania problemu dla tego konkretnego problemu.

2 Epidemia

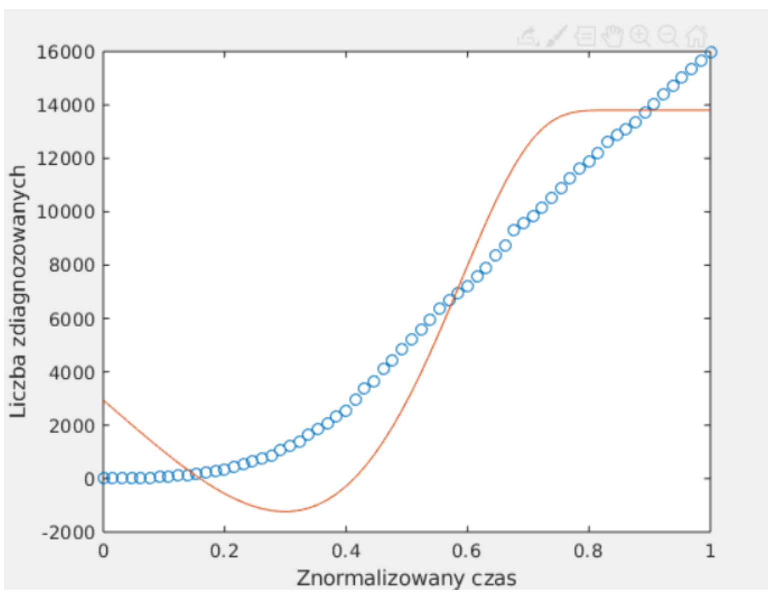
2.1 Porównania algorytmów (czas obliczeń, ew. odległość punktu startowego od rozwiązania, która nie powoduje utraty zbieżności).

Porównania dokonano we wcześniejszym punkcie więc zgodnie z konspektem - ten punkt pomijamy.

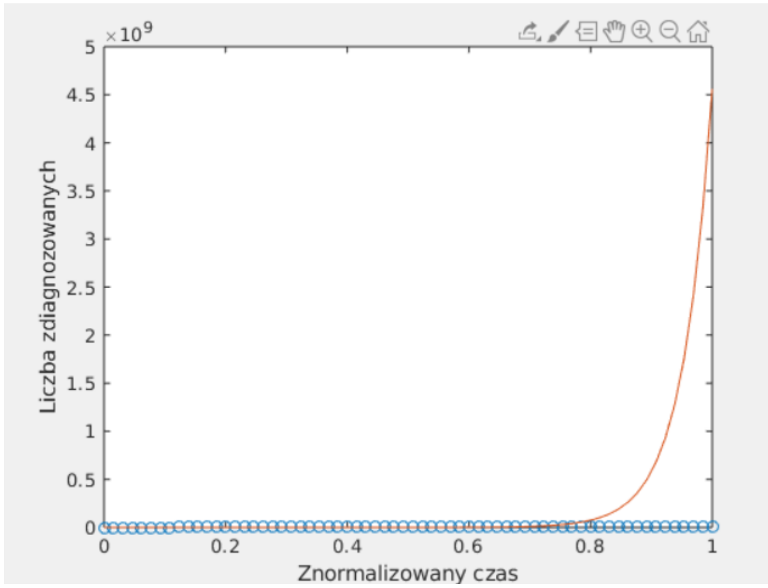
2.2 Porównanie wyników otrzymanych przy stosowaniu róznych norm. Które z 3 algorytmów stosowanych w załączonych skryptach pozwalają na zastosowanie innej normy niż średniokwadratowa?

W każdym z naszych trzech algorytmów można zastosować inną normę niż średniokwadratową, lecz nie w każdej pozwoli to nam uzyskać dobre wyniki.

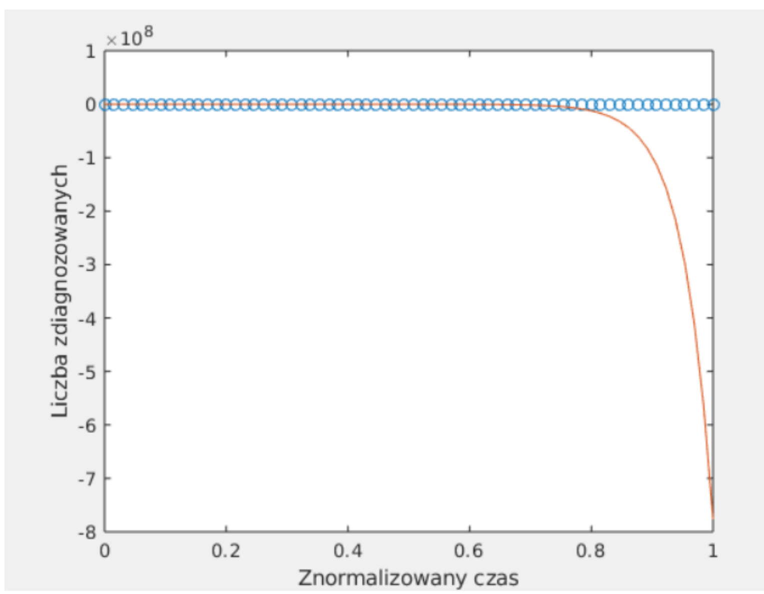
Na poniższych wykresach zastosowaliśmy jako normę maksymalną różnicę między wartością rzeczywistą a przybliżeniem.



Rysunek 1: fminsearch



Rysunek 2: lsqnonlin



Rysunek 3: fminunc

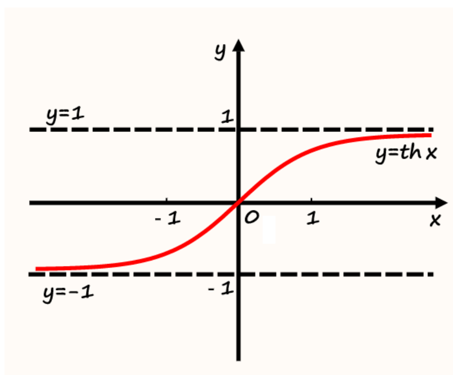
Na podstawie wykresów możemy zauważyć, że żaden z algorytmów przy użyciu tej normy nie przybliża dobrze, choć fminserch poradził sobie zdecydowanie lepiej od pozostałych. Algorytmy lsqnonlin oraz fminunc opierają się na liczeniu pochodnej, więc gdy funkcja nie jest różniczkowalna, nasze algorytmy mogą sobie nie poradzić.

2.3 Inne propozycje funkcji aproksymującej.

W zadaniu możnaby zmienić funkcję opisującą współczynnik beta. W zadaniu wymagane jest aby zmieniała się pomiędzy poziomami - w skrypcie zastosowaliśmy przesunięty arctg. Jednak istnieją inne funkcje także spełniające ten wymóg.

- tangens hiperboliczny

$$f(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$$



- Transponowany sigmoid

$$f(x) = \frac{1}{1 + e^{-x}} - \frac{1}{2}$$