HW6 - Opis ciekawego narzędzia do wizualizacji danych

Mateusz Nizwantowski
01161932@pw.edu.pl
313839

25 grudnia 2022

Streszczenie

Cel pracy domowej: Zapoznanie się z ciekawym narzędziem do wizualizacji danych. Należy stworzyć notatkę na temat jednego wybranego narzędzia/pakietu do wizualizacji danych, który nie pojawił się na zajęciach. Można wybrać narzędzie do wizualizacji w dowolnym języku programowania.

Rozwiązanie powinno zawierać:

- opis narzędzia/pakietu,
- wizualizację przygotowaną w tym narzędziu,
- kod przygotowanej wizualizacji.

1 Wstęp

Dość nietypowym (a przynajmniej na to liczę) narzędziem wybranym do tego zadania domowego (biorąc pod uwagę, że przedmiot nazywa się Techniki wizualizacji danych) jest Matlab. "Gryzie"się on z ideą tego przedmiotu, nie można w nim robić tak ładnych i zjawiskowych wizualizacji jakie robimy na projektach czy laboratoriach, jednak nie taki jest cel Matlaba. W tej pracy chciałbym pokazać, że narzędzia takie jak Matlab czy Excel jak najbardziej są narzędziami do wizualizacji danych, po prostu nie są dla nas (IAD) i ten fakt nie powinien powodować, że traktujemy je jako gorsze.

Musimy odpowiedzieć sobie na pytanie po co robimy wykresy? Uważam że głównym celem wizualizacji jest przekazanie informacji oraz pomoc w zrozumieniu danych z którymi pracujemy w najbardziej wygodny/przejrzysty sposób jak to możliwe. I oczywiście taki cel można osiągnąć na wiele sposobów.

Powinniśmy sobie uświadomić, że Matlab i R są przeznaczone dla dwóch różnych grup ludzi. Wprawdzie obie pracują z danymi jednak mają inne potrzeby, stąd potrzebują innych wizualizacji. Mam o tyle szczęścia, że przez rok studiowałem na MELu (wydział Mechaniki Energetyki i Lotnictwa) i byłem członkiem koła naukowego SKA (Studenckie Koło Astronautyczne). Oba wyżej wymienione środowiska były zupełnie różnie od tego co mogę doświadczyć na MINI (tego skrótu chyba rozwijać nie muszę), przepełnione inżynierami w pełnym tego słowa znaczeniu, programowali mikro-kontrolery w C, robili rysunki techniczne nowych rakiet, symulacje przeprowadzali w Matlabie. Naturalnym dla nich jest używanie wizualizacji w Matlabie, znajomy syntaks, czujniki i urządzenia wykonujące pomiary są wbudowane w ekosystem i przekazują dane w czasie rzeczywistym do środowiska programistycznego, brak potrzeby uczenia się nowego narzędzia, które wymaga innego sposobu myślenia. Wszystkie dane są na miejscu, wystarczy wpisać *plot(x,y)* i mamy wykres, który każdy w kole czy na kierunku rozumie. Nasze wykresy na TWD mają być jasne i czytelne dla każdego, gdyż potencjalne grono odbiorców naszych wizualizacji w przyszłości jest szerokie. Podczas gdy członkowie SKA tworzą wykresy dla inżynierów (często samych siebie), posługują się oni językiem, który każdy w ich organizacji rozumie. Przejdźmy teraz do opisu podstawowego syntaksu wizualizacji wybranego przeze mnie narzędzia.

2 Opis narzędzia

Aby stworzyć jakąkolwiek wizualizację w Matlabie musimy zacząć od słowa kluczowego *figure*. Tworzy ono okno na którym możemy umieścić nasz wykresy. Jeżeli chcemy w jednym oknie umieścić kilka wykresów musimy użyć *subplot(n,m,z)*. Subplot tworzy siatkę rozmiaru $n \times m$ i w z-te pole wpisuje nasz wykres. Jest to bardzo wygodny sposób na rozmieszczanie wykresów i bardzo mi się spodobał.

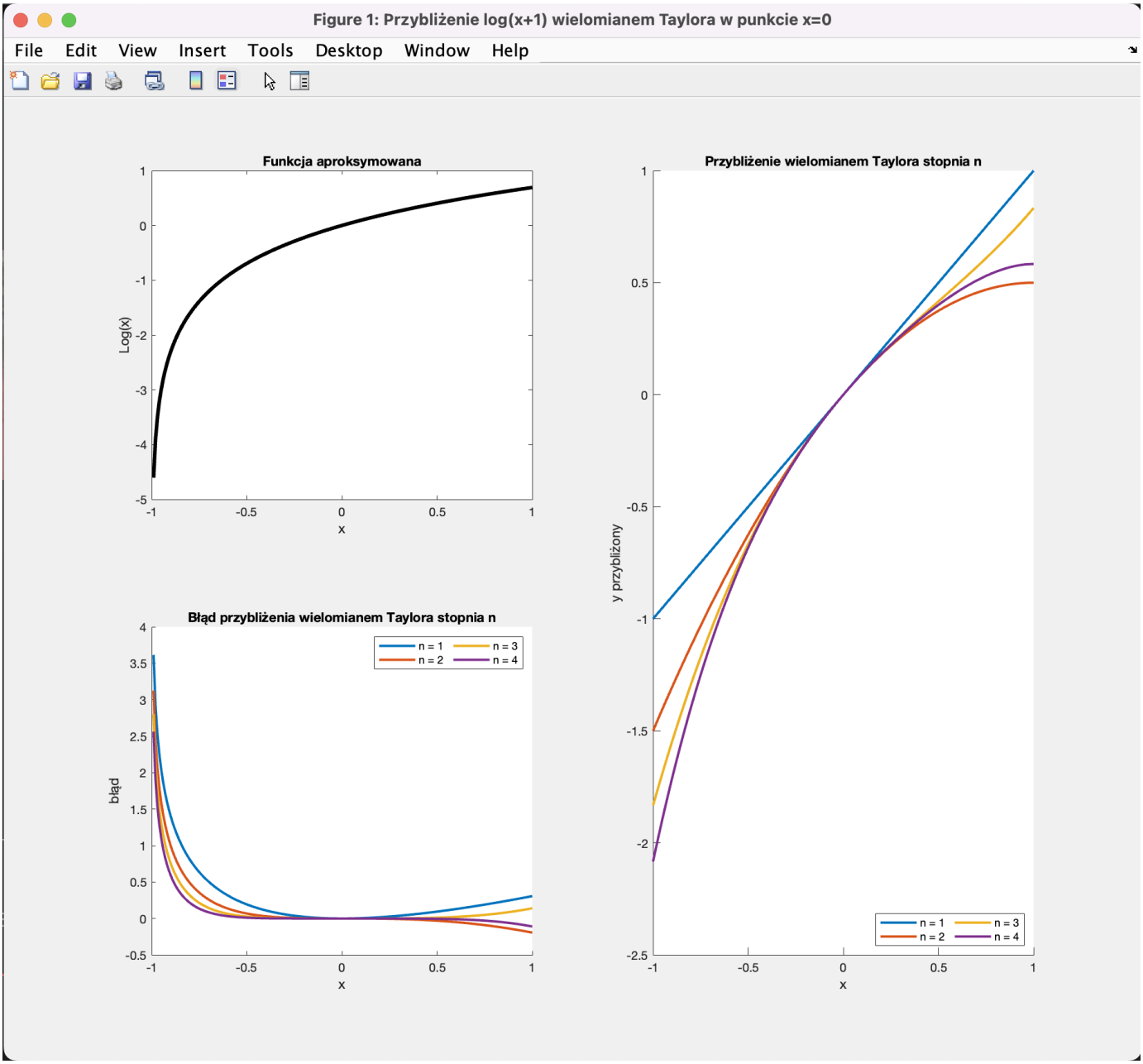
Teraz o najważniejszej funkcji, wspomniałem już o niej wcześniej, mowa o *plot(x,y,...)*, przyjmuje ona wiele argumentów i jest bardzo rozbudowana, dla zainteresowanych polecam oficjalną dokumentację. Znajduje się w niej wiele przykładów a wszystko jest dokładnie i wyczerpująco opisane. Podstawowa wersja rysuje wykres liniowy kolejnych punktów z funkcji f(x) = y. Jednak warto zaznaczyć, że w łatwy sposób da się zmienić podstawowe parametry takie jak kolor linii, styl na np. przerywaną, lub grubość, tak aby dało się odróżnić linie od siebie i aby wizualizacja była przejrzysta.

Kolejnym ciekawym mechanizmem jest sposób rysowania wielu obiektów (np. linii) na jednym wykresie. Robi się to w pętli przy użyciu słów kluczowych *hold on* i *hold off*. Bez ich użycia kolejne wywołanie plota nadpisało by poprzedni. Trzeba przyznać, że to nietypowe rozwiązanie i znacząco odmienne podejście niż to, do którego przywykliśmy pracując z ggplot2. Przy używaniu pętli należy pamiętać, że oprócz indentacji używa się słowa kluczowego *end* do zasygnalizowania, że tu kończy się blok kodu.

W naszej dyspozycji mamy też funkcję, których nazwa jest samo tłumacząca się: *xlabel('Nazwa osi x'), ylabel('Nazwa osi y'), title('Tytuł wykresu')*. Pomagają one nadać sensu danym przedstawionym na wykresie. Możemy też dodać legendę, ustawić gdzie się znajduje i jakie informacje są tam zawarte.

Warto zaznaczyć, że otrzymane wizualizacje są interaktywne. Łatwo można przybliżać, sprawdzać wartości, rysować po nich. Jeżeli tak się zdarzyło, że legenda zasłania jakąś ważną część wykresu, możemy na nią kliknąć i przytrzymując przycisk myszy przesunąć ją w bardziej odpowiednie miejsce. Matlab daje też możliwość eksportowania ich z tego poziomu do pliku formatów pdf, jpeg, png i wiele innych.

3 Wizualizacja



Rysunek 1: Okno wizualizacji w Matlabie

4 Kod wizualizacji

Listing 1: Właściwy kod wizualizacji

```
figure('Name',
        'Przyblizenie log(x+1) wielomianem Taylora w punkcie x=0')
x = -1:0.01:1;
y = log(x+1);
n = 4;
opisy = ['n = 1' 'n = 2' 'n = 3' 'n = 4'];

%lewy gorny wykres
subplot(2,2,1)
plot(x,y, LineWidth=3,Color='black')
title('Funkcja aproksymowana')
xlabel('x')
ylabel('Log(x)')

%prawy wykres
subplot(2,2,[2 4])
A = zeros([length(x) n]);
for i = 1:length(x)
    A(i,:) = szeregLn(x(i),n);
end

for i = 1:n
    hold on
    plot(x,A(:,i), LineWidth=2)
    hold off
end
title('Przyblizenie wielomianem Taylora stopnia n')
xlabel('x')
ylabel('y przyblizony')
lgd = legend(opisy);
lgd.NumColumns = 2;
lgd.Location = 'southeast';

%lewy dolny wykres
subplot(2,2,3)
for i = 1:n
    hold on
    plot(x, A(:,i)'-y, LineWidth=2)
    hold off
end
lgd = legend(opisy);
lgd.NumColumns = 2;
```

```
43 title('Blad przyblizenia wielomianem Taylora stopnia n')
44 xlabel('x')
45 ylabel('blad')
```

Listing 2: Funkcja pomocnicza do wizualizacji

```
1  function [yaprox] = szeregLn(x, Nmax)
2  %SZEREGLN Funkcja obliczaj ca szereg Maclaurina funkcji
3  %   ln(x+1) w punkcie x stopnia Nmax
4  if nargin == 1
5      Nmax = 10;
6  end
7  A = zeros(1,Nmax);
8  for i = 1:Nmax
9      A(:,i) = ((-1)^(i+1))*(x.^(i))/(i);
10 end
11 yaprox = cumsum(A);
12 end
```

5 Podsumowanie

Nie mamy tu tak dużo możliwości modyfikowania wykresów jak w R. Jednak w Matlabie można szybko w łatwy sposób tworzyć przejrzyste (nie zawsze najładniejsze) wizualizacje, które zostaną bez problemu zrozumiane przez naszych współpracowników - specjalistów. Biorąc to wszystko pod uwagę, stwierdzeniem nienaukowym jest, że Matlab robi brzydkie wizualizacje dlatego nie powinno się go używać do tworzenia wykresów. Matlab jako narzędzie spełnia swoje zadanie, nie jest po prostu odpowiedni dla nas. Ale jest grupa ludzi, dla których Matlab robi dokładnie to czego oczekują i potrzebują.