Tomasz Bocheński

MNUM Projekt 2 – DOKUMENTACJA cz. 1

Zadanie 1

Treść:

Proszę napisać program służący do obliczania wartości własnych macierzy nieosobliwych metodą rozkładu QR w dwóch wersjach: bez przesunięć i z przesunięciami dla macierzy symetrycznej, oraz w wersji z przesunięciami dla macierzy niesymetrycznej. Następnie proszę przetestować skuteczność (zbieżność) obu wersji algorytmu dla 30 różnych macierzy losowych o wymiarach: 5x5, 10x10, 20x20. Proszę podać średnią liczbę iteracji dla metody bez przesunięć i z przesunięciami. Dla wybranych macierzy proszę porównać otrzymane wyniki z wartościami własnymi obliczonymi poleceniem eig.

Uwagi:

- W programie nie można wykorzystać dostępnych w Matlabie poleceń qr i eig.

- Macierz $B = (A+A^T)$ jest macierzą symetryczną dla dowolnej macierzy A.

Krótki opis zastosowanych algorytmów:

W celu wyznaczenia wartości własnych macierzy można wykorzystać metodę QR znajdowania wartości własnych. Istnieją dwa algorytmy dla tej metody.

Pierwszy to algorytm podstawowy, czyli metoda QR bez przesunięć. Za pomocą tego algorytmu można wyznaczać wartości własne macierzy symetrycznych. Idea pojedynczego kroku tej metody to:

$A^{(k)} = Q^{(k)} * R^{(k)}$ (faktoryzacja)

$A^{(k+1)} = R^{(k)} * Q^{(k)}$;

Macierz $A^{(k+1)}$ jest przekształconą przez podobieństwo macierzą $A^{(k)}$, czyli posiada te same wartości własne. Ponadto dla k->∞ macierz $A^{(k)}$ zbiega do macierzy diagonalnej z wartościami własnymi na diagonali. Zbieżność elementu poddiagonalnego $a_{i+1,i}$ do zera jest liniowa z ilorazem zbieżności $| \lambda_{i+1} / \lambda_i |$. Zatem metoda ta, chociaż prawidłowa, dla wartości własnych leżących blisko siebie jest wolno zbieżna.

Drugi algorytm stosuje się właśnie w celu poprawienia tej zbieżności. Jest to algorytm metody QR z przesunięciami. Można go stosować zarówno dla macierzy symetrycznych jak i niesymetrycznych. Idea pojedynczego kroku tej metody to:

$A^{(k)} – p_kI = Q^{(k)} * R^{(k)}$

$A^{(k+1)} = R^{(k)} * Q^{(k)} + p_kI$

W tym przypadku, jak łatwo się domyśleć, zbieżność jest liniowa z ilorazem $| (\lambda_{i+1} - p_k) / (\lambda_i - p_k) |$.

Za p_k najlepiej przyjąć wartość własną podmacierzy 2x2 z prawego dolnego rogu bliższą wartości w prawym dolnym rogu. Gdy nastąpi wyzerowanie wszystkich elementów ostatniego wiersza (oprócz elementu na diagonalnej) otrzymuje się jedną z wartości własnych. Następnie należy usunąć ostatni wiersz i ostatnią kolumnę i powtórzyć operacje.

W celu wyznaczenia rozkładu QR stosowany jest zmodyfikowany algorytm Gram'a Schmidt'a. Ma on znacznie lepsze własności numeryczne niż postać niezmodyfikowana. Algorytm ten przeprowadza proces ortogonalizacji w innej kolejności. W standardowym algorytmie ortogonalizuje się wszystkie kolumny macierzy po kolei. Algorytm zmodyfikowany po wyznaczeniu kolejnej kolumny ortogonalnej, ortogonalizuje wobec niej wszystkie następne kolumny.

Prezentacja otrzymanych wyników:

Otrzymane wyniki zostaną zaprezentowane w postaci tabelek. Taka forma prezentacji wyników ułatwi ich analizowanie oraz porównywanie.

STATYSTYKI ORAZ PORÓWNANIE WYNIKÓW DLA LOSOWYCH MACIERZY

MACIERZ SYMETRYCZNA BEZ PRZESUNIĘĆ

		PRECYZJA	
		0.1	0.01
MACIERZE 5X5	Średnia ilość iteracji	132.6333	29053.03
	Średni czas wykonania	0.0128365	2.94873
MACIERZE 10X10	Średnia ilość iteracji	475.5333	2116.067
	Średni czas wykonania	0.11624	0.546547
MACIERZE 20X20	Średnia ilość iteracji	1072	3307.933
	Średni czas wykonania	0.807093	2.67896

Wynik zaimplementowanej funkcji	Wynik wywołania eig
8.123140	-6.804449
-6.804449	-4.604427
6.414907	-3.459168
-4.604427	-2.414454
-3.459168	1.655185
-2.414454	6.414907
1.655185	8.123140

MACIERZ SYMETRYCZNA Z PRZESUNIĘCIAMI

		PRECYZJA	
		0.00001	0.0000001
MACIERZE 5X5	Średnia ilość iteracji	8.166667	8.833333
	Średni czas wykonania	0.00118332	0.00124838
MACIERZE 10X10	Średnia ilość iteracji	16.33333	18.70000
	Średni czas wykonania	0.00349458	0.00400227
MACIERZE 20X20	Średnia ilość iteracji	31.86667	34.96667
	Średni czas wykonania	0.0138762	0.0150144

Wynik zaimplementowanej funkcji	Wynik wywołania eig
7.896881	-3.711954
4.033532	-2.908275
2.288039	0.207537
1.370253	1.370253
0.207537	2.288039
-2.908275	4.033532
-3.711954	7.896881

MACIERZ NIESYMETRYCZNA Z PRZESUNIĘCIAMI

		PRECYZJA	
		0.00001	0.0000001
MACIERZE 5X5	Średnia ilość iteracji	10.70000	11.93333
	Średni czas wykonania	0.00197396	0.00227961
MACIERZE 10X10	Średnia ilość iteracji	23.80000	26.60000
	Średni czas wykonania	0.00715347	0.00795967
MACIERZE 20X20	Średnia ilość iteracji	49.63333	54.50000
	Średni czas wykonania	0.0321577	0.0357305

Wynik zaimplementowanej funkcji	Wynik wywołania eig
-3.104895	-3.104895
-0.387191	-0.387195
-0.387193	-0.387195
0.611590	0.611596
0.611596	0.611596
2.559588	2.184513
2.184514	2.55958

Komentarz do otrzymanych wyników oraz wnioski z eksperymentów (ocena poprawności wyników, dokładności, efektywności algorytmów itd.):

Średnia liczba iteracji potrzebna do wyznaczenia wartości własnych macierzy symetrycznej za pomocą metody rozkładu QR bez przesunięć jest zdecydowanie większa od średniej liczby iteracji potrzebnej do wyznaczenia wartości własnych macierzy symetrycznej i niesymetrycznej za pomocą metody rozkładu QR z przesunięciami. Podobnie jest z czasem wykonania. Zatem wyniki te zgadzają się z teoretycznymi założeniami. Szybkość zbieżności w algorytmie rozkładu QR z przesunięciami jest poprawiana, dlatego aby otrzymać konkretny wynik potrzebna jest mniejsza liczba iteracji, a co za tym idzie, również mniejszy czas. Algorytm do wyznaczania wartości własnych dla macierzy symetrycznych jak i niesymetrycznych jest taki sam. Średnia liczba iteracji oraz średni czas wykonania jest podobny w obu tych przypadkach, minimalnie lepsze wyniki osiągają macierze symetryczne. W przypadku macierzy niesymetrycznych mogą pojawić się wartości własne zespolone, należy więc używać arytmetyki zespolonej. W ogólnym przypadku wraz ze wzrostem stopnia macierzy kwadratowej wzrasta też średnia liczba iteracji oraz średni czas. Widać to dobrze dla macierzy symetrycznych i niesymetrycznych rozwiązywanych metodą rozkładu QR z przesunięciami. W przypadku macierzy symetrycznych rozwiązywanych za pomocą metody QR bez przesunięć nie zawsze da się zaobserwować taką własność. Wynika to z faktu, że w metodzie z przesunięciami zawsze dobierane jest optymalne przesunięcie, aby zapewnić największą zbieżność. W metodzie bez przesunięć operacje takie nie są wykonywane, dlatego na ostateczny czas i ostateczną liczbę iteracji ma przede wszystkim wpływ zbieżność losowo wygenerowanych macierzy. Może się zdarzy, że losowo wygenerowane macierze 5x5 będą miały wartości własne leżące obok siebie. Wtedy metoda te będzie wolno zbieżna i potrzebna będzie duża liczba iteracji i długi czas aby wyznaczyć te wartości. W przypadku wygenerowania losowych macierzy 10x10 dobrze zbieżnych może się okazać że chociaż stopień macierzy kwadratowej jest większy, to zarówno liczba iteracji jak i czas potrzebny do wyznaczenia wartości własnych jest mniejszy (w porównaniu do macierzy kwadratowych stopnia 5). Ponadto wraz ze wzrostem precyzji (czyli dopuszczalnego odchylenia od zera wartości leżących pod diagonalną) wzrasta liczba iteracji oraz czas wykonywania. Jest to sprawa oczywista, ponieważ ze wzrostem precyzji wiąże się wzrost iteracji, a co za tym idzie, wzrost czasu wykonania. Dowodem na to, że zaimplementowane metody poprawnie wyznaczają wartości własne jest porównanie wartości wyznaczonych za pomocą tych metod z wartościami wyznaczonymi za pomocą polecenia eig. Można zauważyć, że odpowiednie wartości własne zgadają się. Wartości własne otrzymywane w przypadku metody z przesunięciami dla macierzy niesymetrycznych minimalnie różnią się od wartości własnych wyznaczonych przez eig. Jednak wynika to z przyjętej dokładności, precyzji. Kolejność wyznaczania wartości własnych zależy od algorytmu.

Zadanie 2

Treść:

Dla następujących danych pomiarowych (próbek):

x_1 = -5, x_2 = -4, x_3 = -3, x_4 = -2, x_5 = -1, x_6 = 0, x_7 = 1, x_8 = 2, x_9 = 3, x_{10} = 4, x_{11} = 5,

y_1 = 63,6802, y_2 = 33,2744, y_3 = 16,1215, y_4 = 4,7061, y_5 = 0,2707, y_6 = -0,1198, y_7 = -0,0597, y_8 = -0,0080, y_9 = 3,4085, y_{10} = 12,0457, y_{11} = 25,2401,

metodą najmniejszych kwadratów należy wyznaczyć funkcje wielomianową y = f(x) najlepiej aproksymującą te dane (proszę przetestować wielomiany różnych rzędów). W sprawozdaniu proszę przedstawić na rysunku otrzymaną funkcję na tle danych. Do rozwiązania zadania najmniejszych kwadratów proszę wykorzystać:

a) układ równań normalnych;

b) układ równań liniowych z macierzą R wynikającą z rozkładu QR macierzy układu równań problemu.

Dla każdego układu równań proszę obliczyć błąd rozwiązania jako normę residuum (wektor residuum r = Ax-b).

Uwagi:

- Rysowaną funkcję proszę próbkować 10 razy częściej niż dane.

- Dane są obarczone pewnym błędem (szumem pomiarowym).

Krótki opis zastosowanych algorytmów:

LZNK (liniowe zadanie najmniejszych kwadratów) to zadanie polegające na rozwiązaniu układu m równań liniowych z n zmiennymi, gdzie n < m, w sensie minimalizacji normy drugiej wektora niespełniania równań.

W podanym zadaniu wykorzystywane są dwie metody rozwiązywania LNZK. Obie one odnoszą się do macierzy o pełnym rzędzie. Gdy macierz jest niepełnego rzędu używa się algorytmu opartego na rozkładzie SVD.

Dla macierzy A, będącej macierzą pełnego rzędu, symetryczna i kwadratowa macierz A^TA jest dodatnio określona, wszystkie jej wartości własne są dodatnie. Układ równań wynikający dla takiej macierzy, przedstawiony w postaci $A^TAx = A^Tb$, nazywany jest właśnie układem równań normalnych i ma on w tym przypadku jednoznaczne rozwiązanie.

Jednak w przypadku, gdy macierz A jest słabo uwarunkowana, macierz A^TA staje się jeszcze słabiej uwarunkowana. Jest to zachowanie niekorzystne i w takim przypadku należy skorzystać z rozkładu QR (zalecany rozkład wąski). W wyniku rozkładu macierzy A na macierze Q i R otrzymuje się układ równań postaci $Rx = Q^Tb$. Metoda otrzymywania rozkładu wąskiego QR zastosowana w tym zadaniu to zmodyfikowany algorytm Gram'a Schmidt'a.

Zmodyfikowany algorytm Gram'a Schmidt'a ma znacznie lepsze własności numeryczne niż postać niezmodyfikowana. Algorytm ten przeprowadza proces ortogonalizacji w innej kolejności. W standardowym algorytmie ortogonalizuje się wszystkie kolumny macierzy po kolei. Algorytm zmodyfikowany po wyznaczeniu kolejnej kolumny ortogonalnej, ortogonalizuje wobec niej wszystkie następne kolumny.

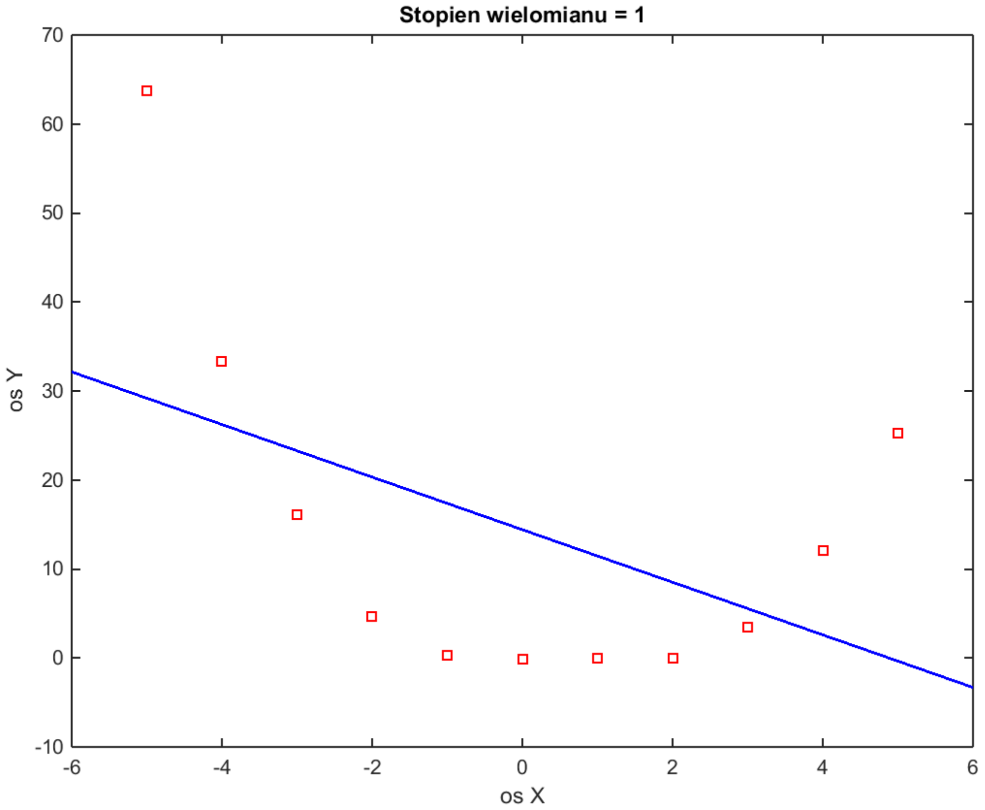
Prezentacja otrzymanych wyników:

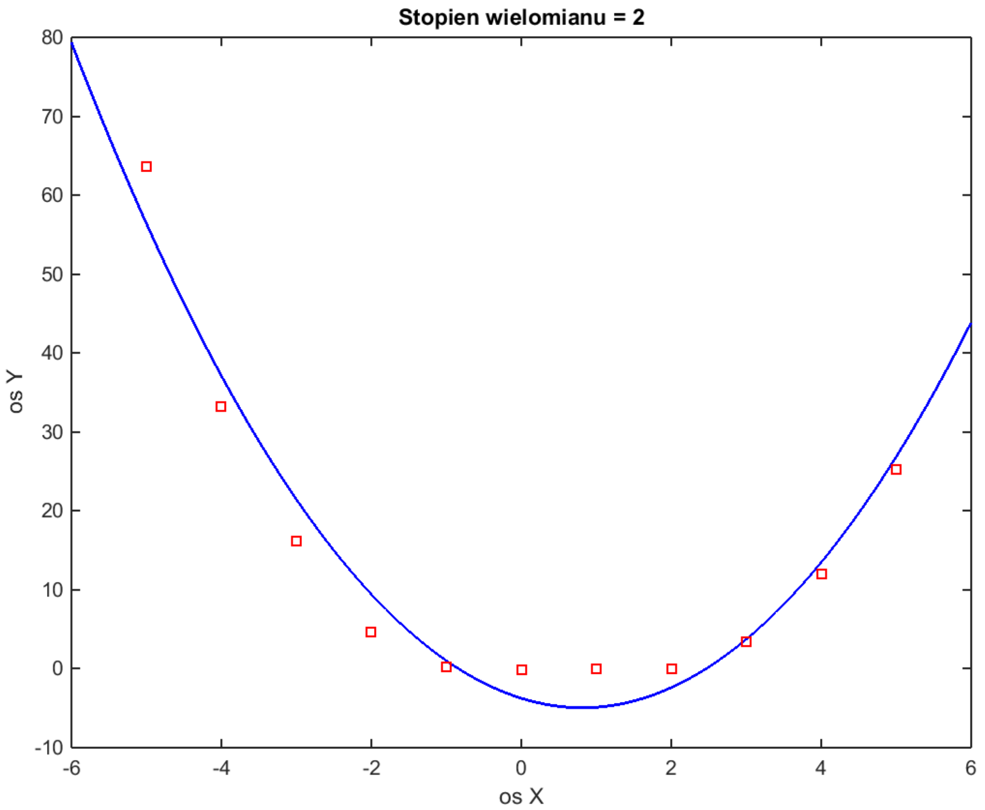
Otrzymane wyniki prezentowane są w postaci tabelki, która ułatwia analizę i porównanie wyników, oraz rysunków z obliczonymi funkcjami na tle danych.

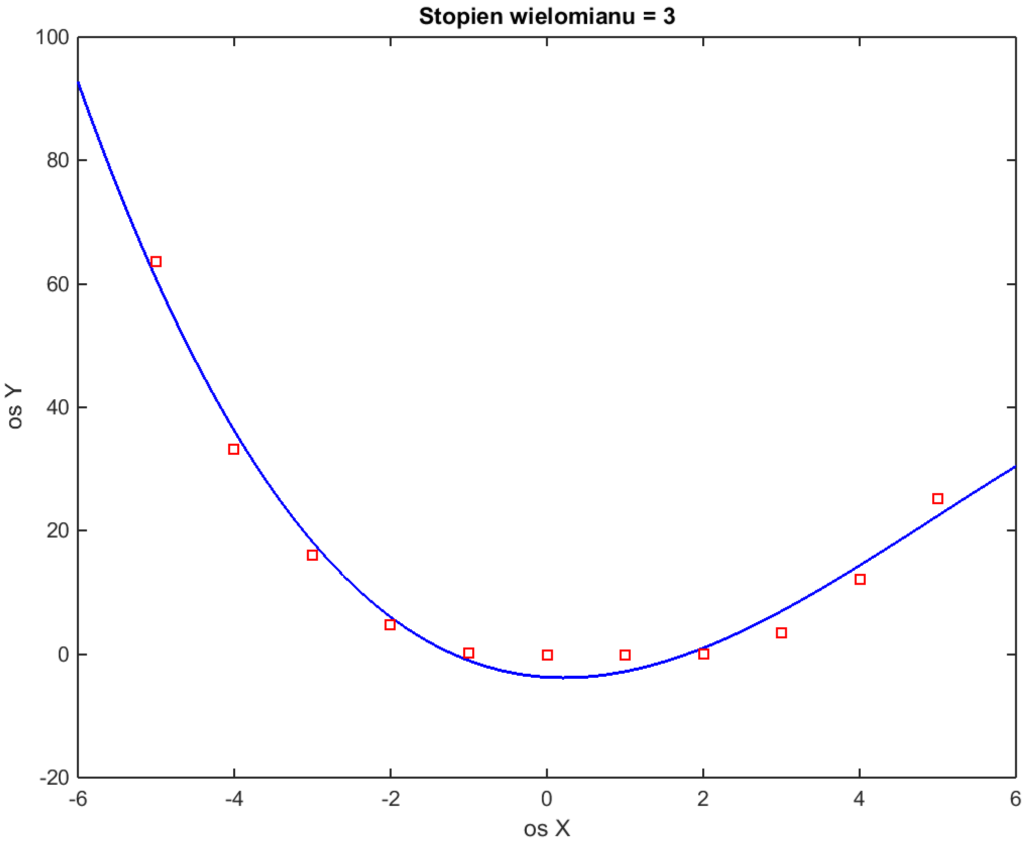
TABELKA:

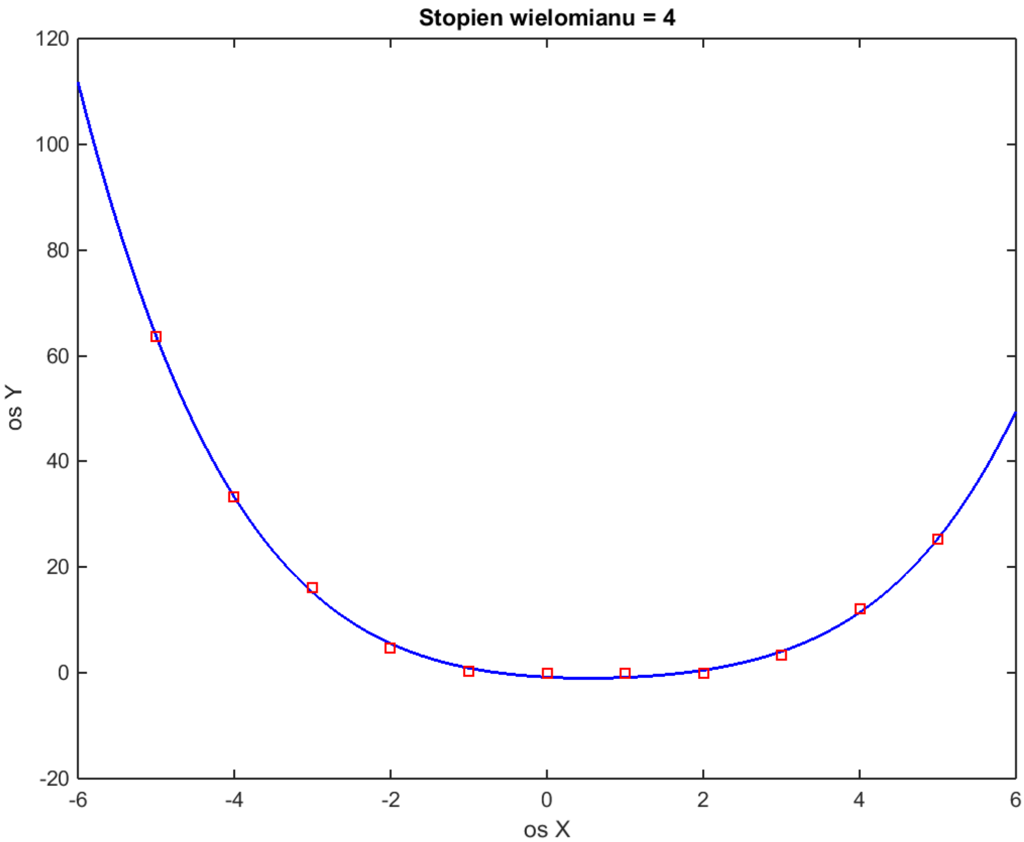
STOPIEŃ WIELOMIANU:		UKŁ ROWNAN NORMALNYCH	UKŁ ROWNAN Z MACIERZA R
1	Błąd rozwiązania	54.70292	54.70292
	Czas rozwiazania	0.000206154	0.00210036
2	Błąd rozwiązania	12.87770	12.87770
	Czas rozwiazania	0.000208385	0.00123514
3	Błąd rozwiązania	8.509205	8.509205
	Czas rozwiazania	0.000158408	0.00141229
4	Błąd rozwiązania	2.005876	2.005876
	Czas rozwiazania	0.000171795	0.0012869
5	Błąd rozwiązania	1.971971	1.971971
	Czas rozwiazania	0.000172241	0.00156356
6	Błąd rozwiązania	1.711258	1.711258
	Czas rozwiazania	0.00019009	0.00157917
7	Błąd rozwiązania	0.8184626	0.8184626
	Czas rozwiazania	0.000162871	0.00147922
8	Błąd rozwiązania	0.05910573	0.05910573
	Czas rozwiazania	0.000181612	0.00161889
9	Błąd rozwiązania	0.05885292	0.05885292
	Czas rozwiazania	0.000279334	0.00175275
10	Błąd rozwiązania	2.486854e-11	4.766269e-11
	Czas rozwiazania	0.000180273	0.00293702

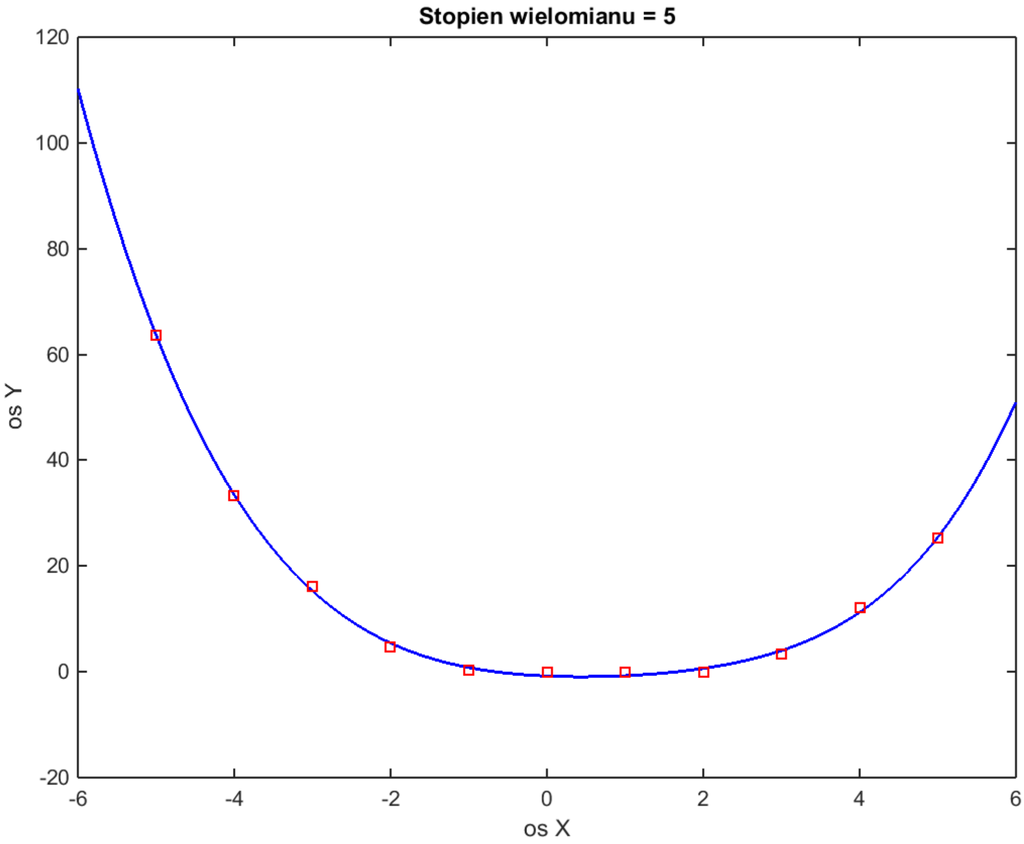
RYSUNKI DLA UKŁ RÓWNAŃ NORMALNYCH:

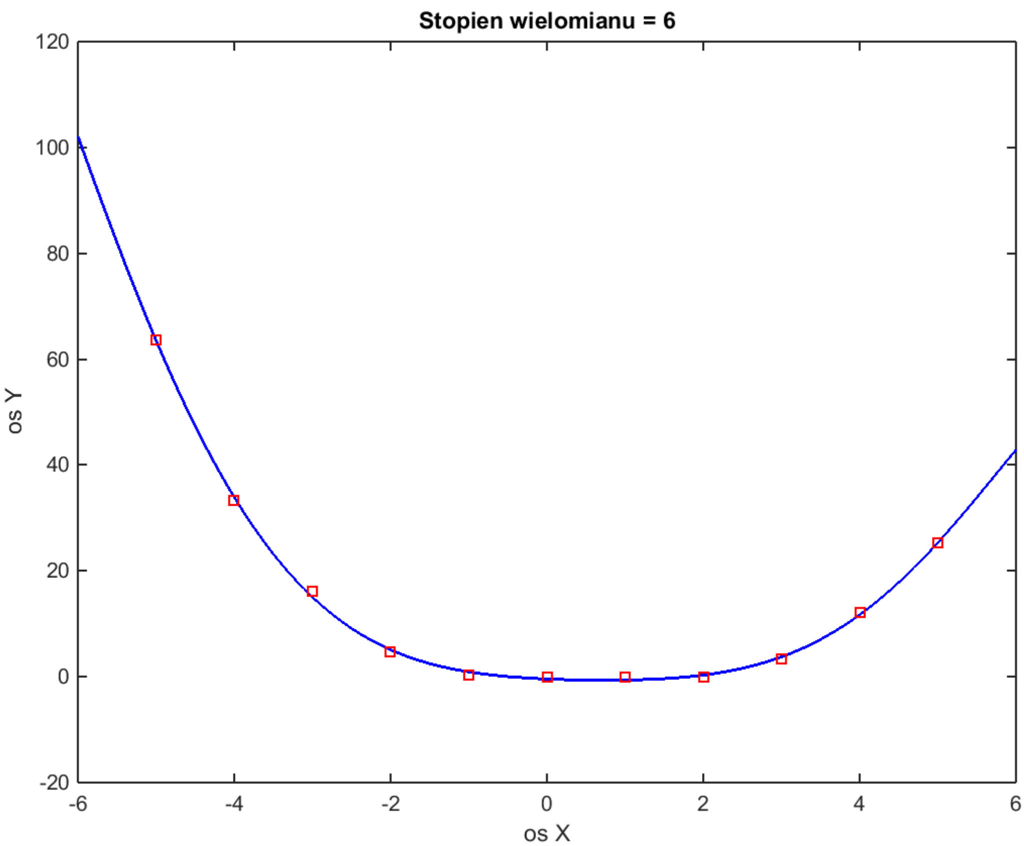


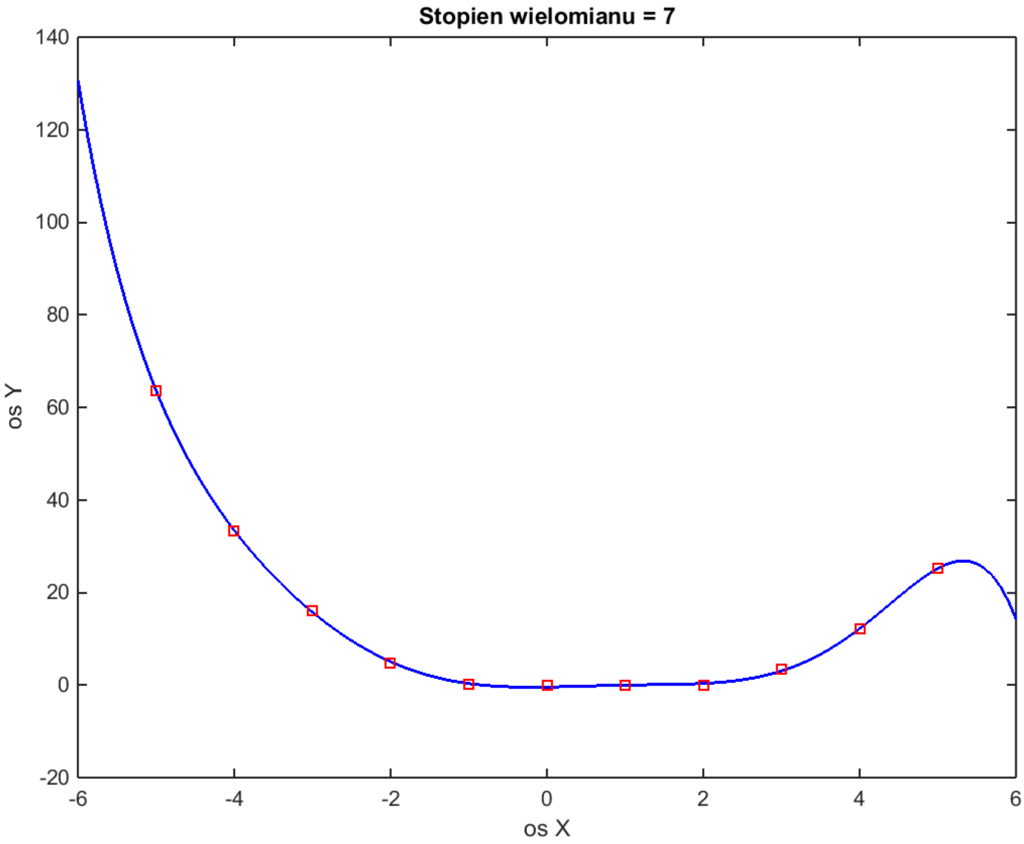


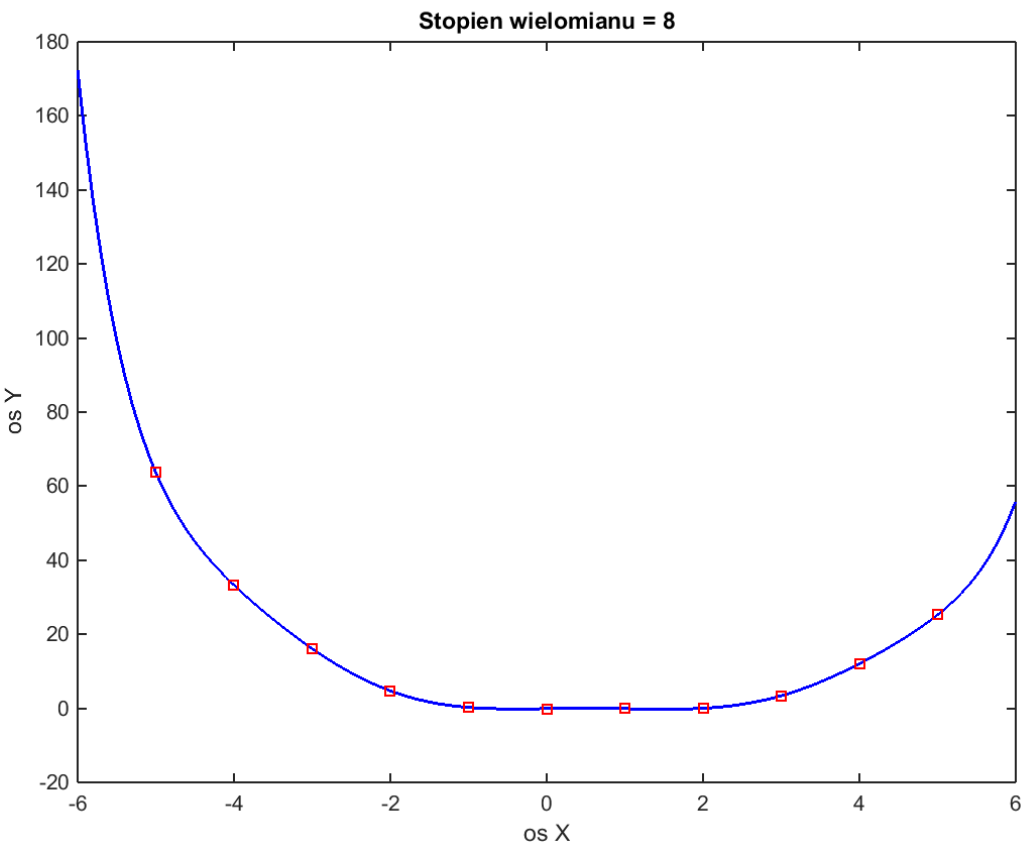

Stopien wielomianu = 3
100
80
60
40
20
0
-20
-6
-4
-2
0
2
4
6
os Y
os X

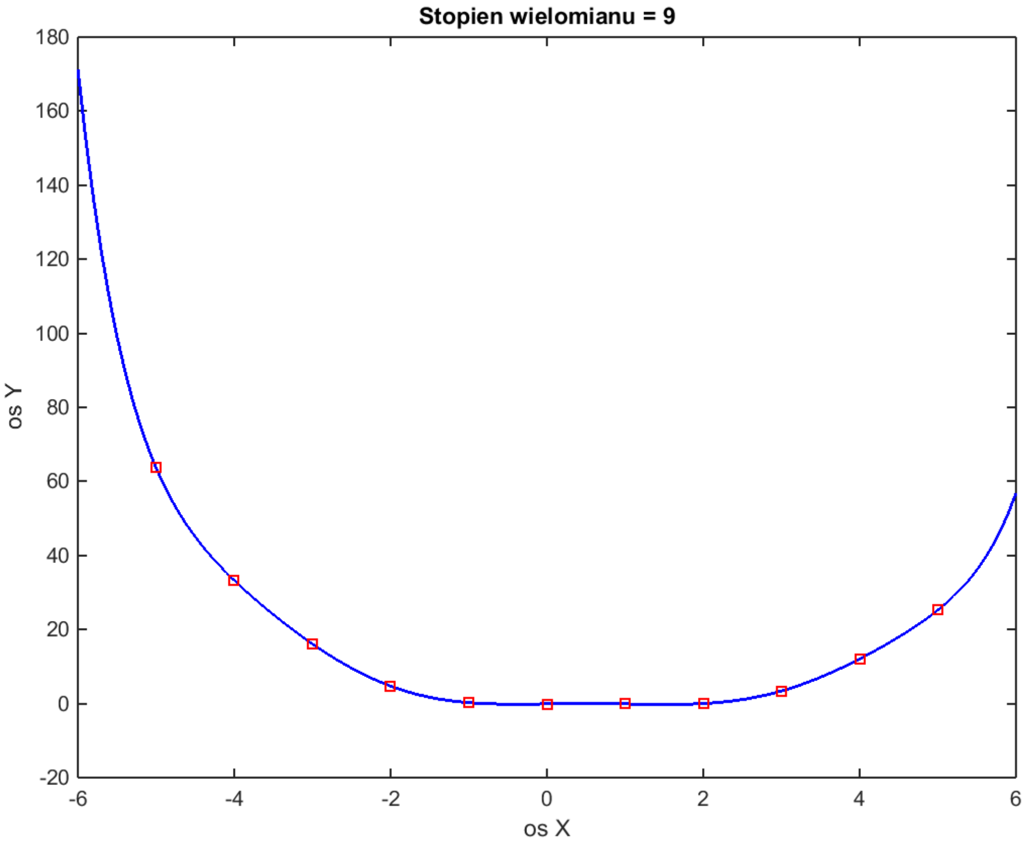

Stopien wielomianu = 4
120
100
80
60
40
20
0
-20
-6
-4
-2
0
2
4
6
os Y
os X

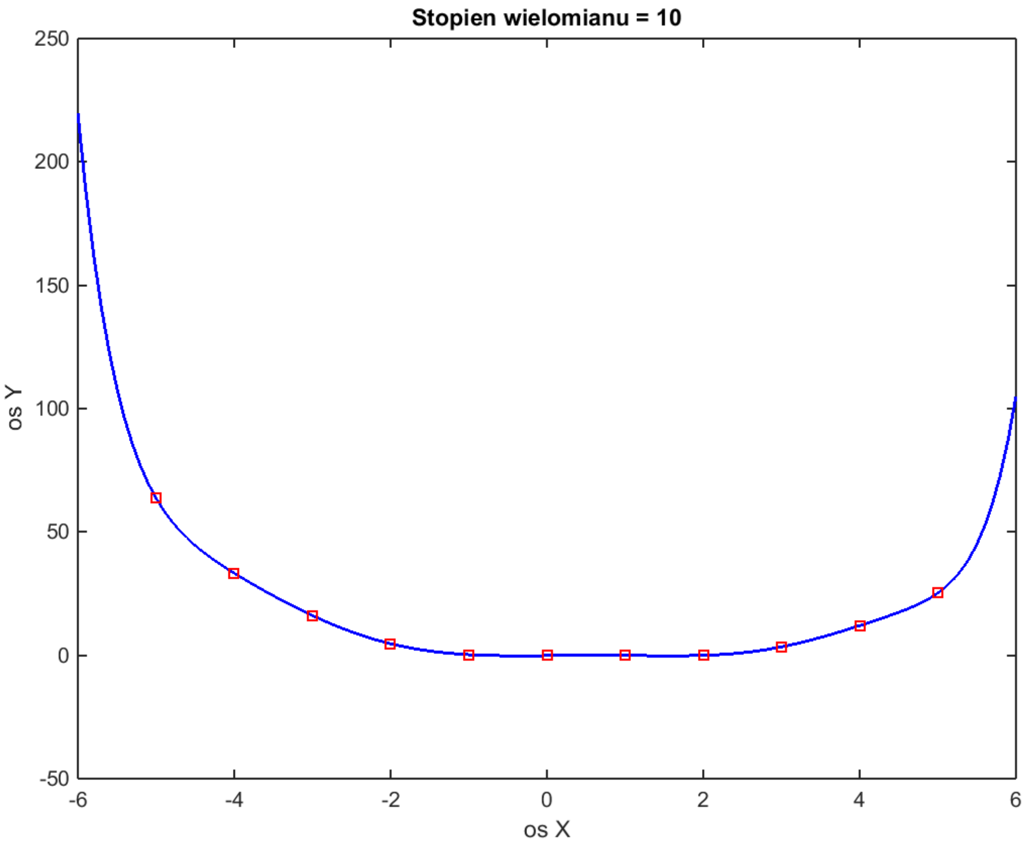

Stopien wielomianu = 5
os Y
os X
120
100
80
60
40
20
0
-20
-6
-4
-2
0
2
4
6


Stopien wielomianu = 6
os Y
os X
120
100
80
60
40
20
0
-20
-6
-4
-2
0
2
4
6

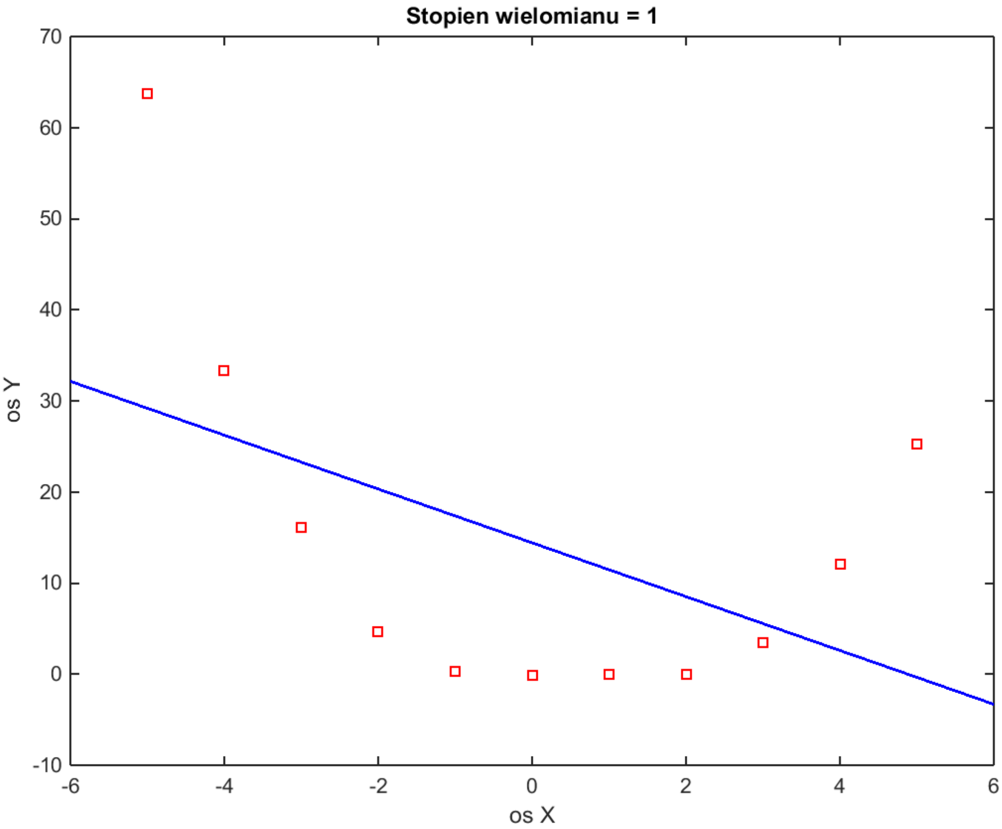

Stopien wielomianu = 7
140
120
100
80
60
40
20
0
-20
os Y
-6
-4
-2
0
2
4
6
os X

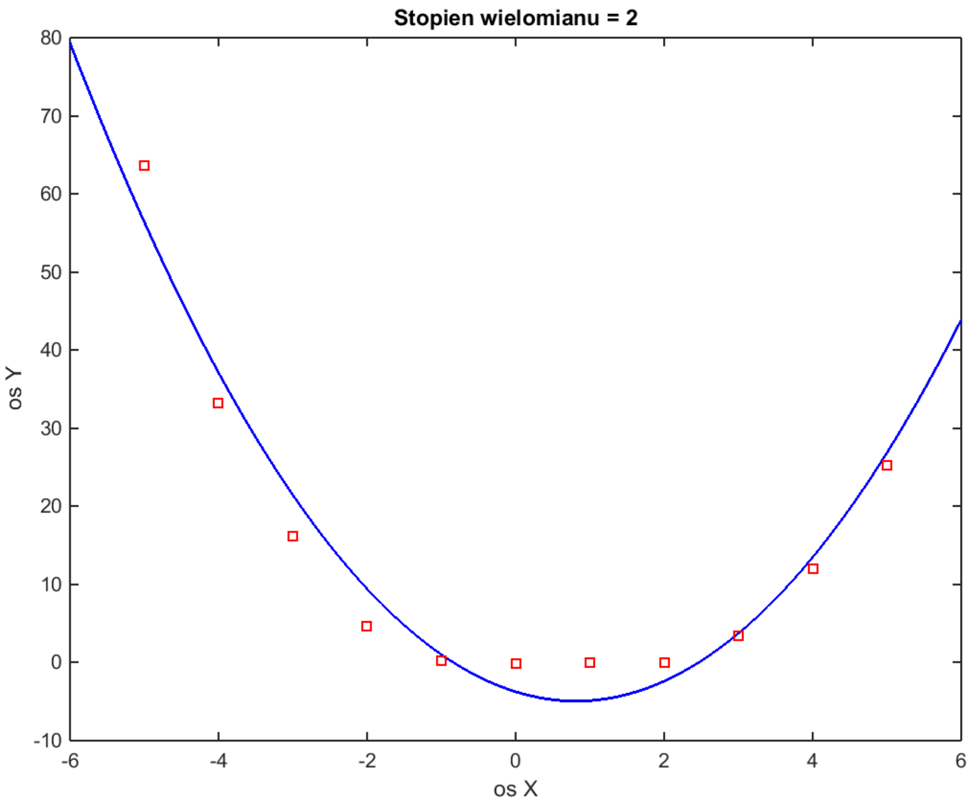

Stopien wielomianu = 8
180
160
140
120
100
80
60
40
20
0
-20
os Y
-6
-4
-2
0
2
4
6
os X

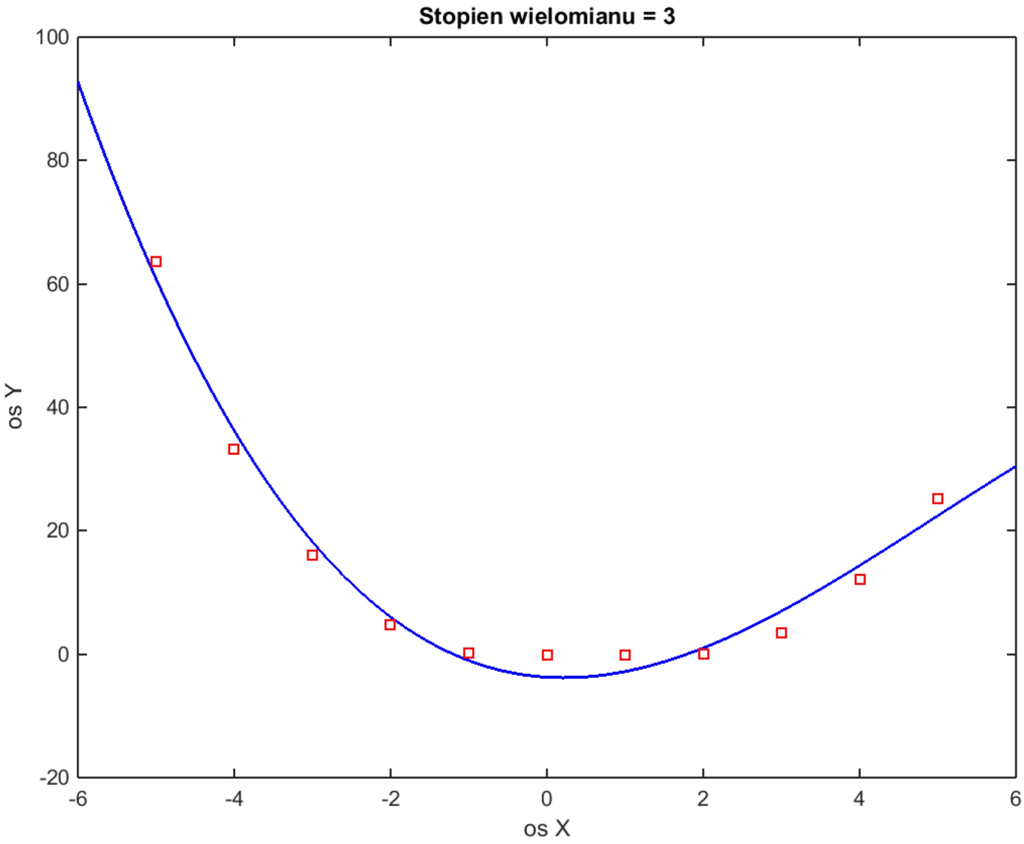

Stopien wielomianu = 9
os Y
os X
180
160
140
120
100
80
60
40
20
0
-20
-6
-4
-2
0
2
4
6

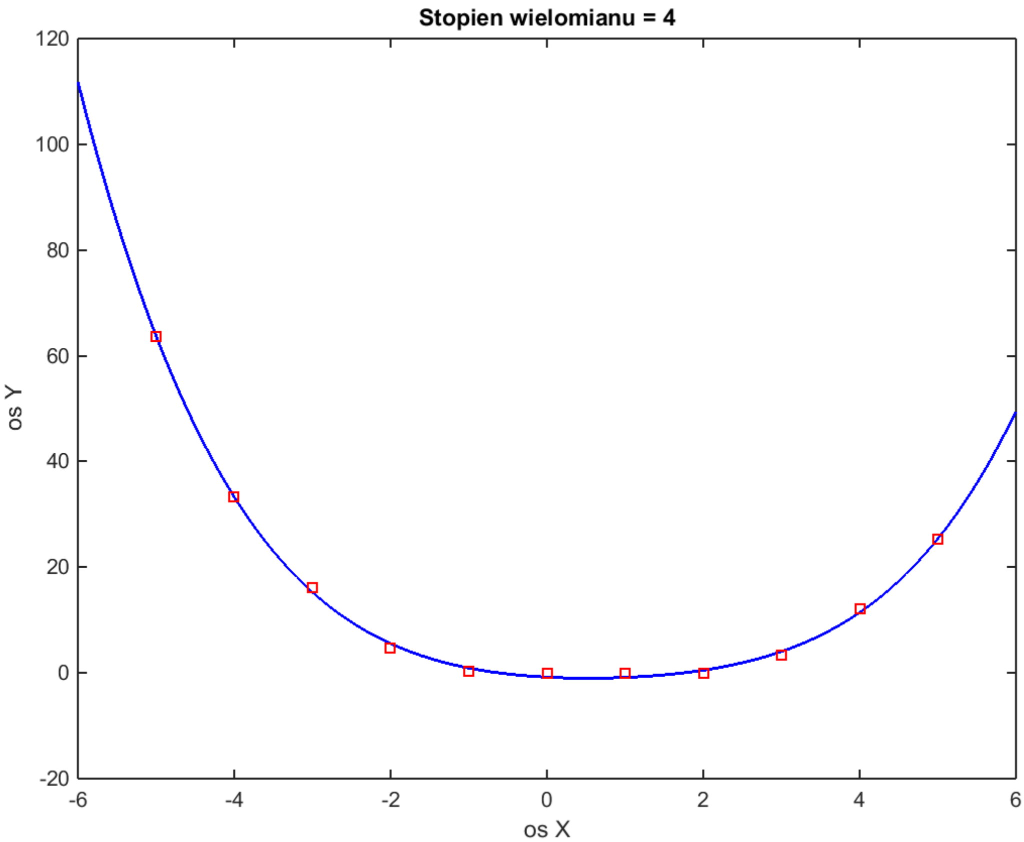

Stopien wielomianu = 10
os Y
os X
250
200
150
100
50
0
-50
-6
-4
-2
0
2
4
6

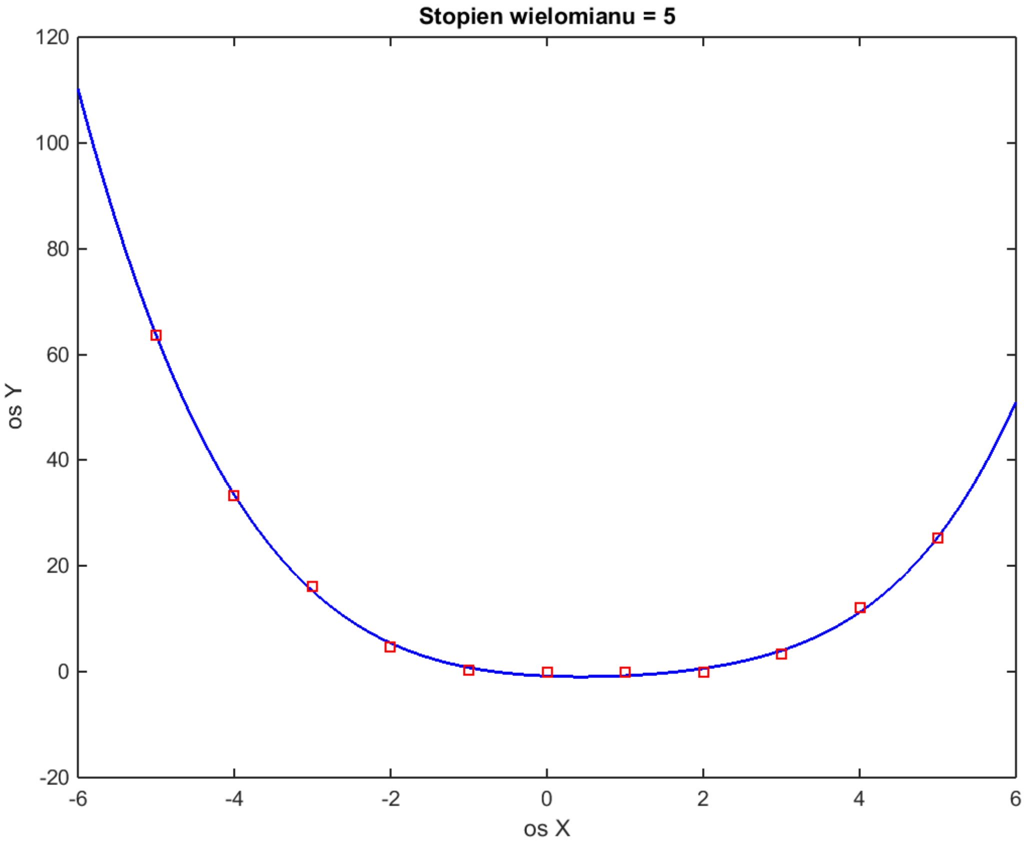
RYSUNKI DLA UKŁ RÓWNAŃ Z MACIERZĄ R:

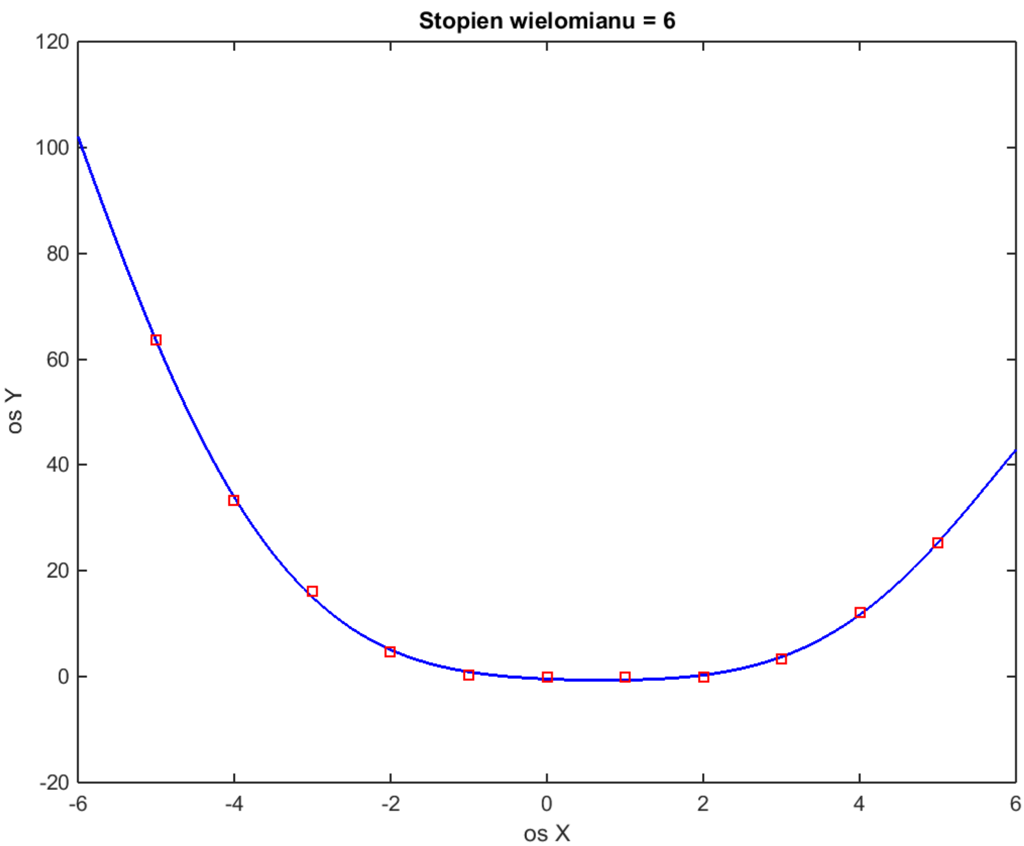


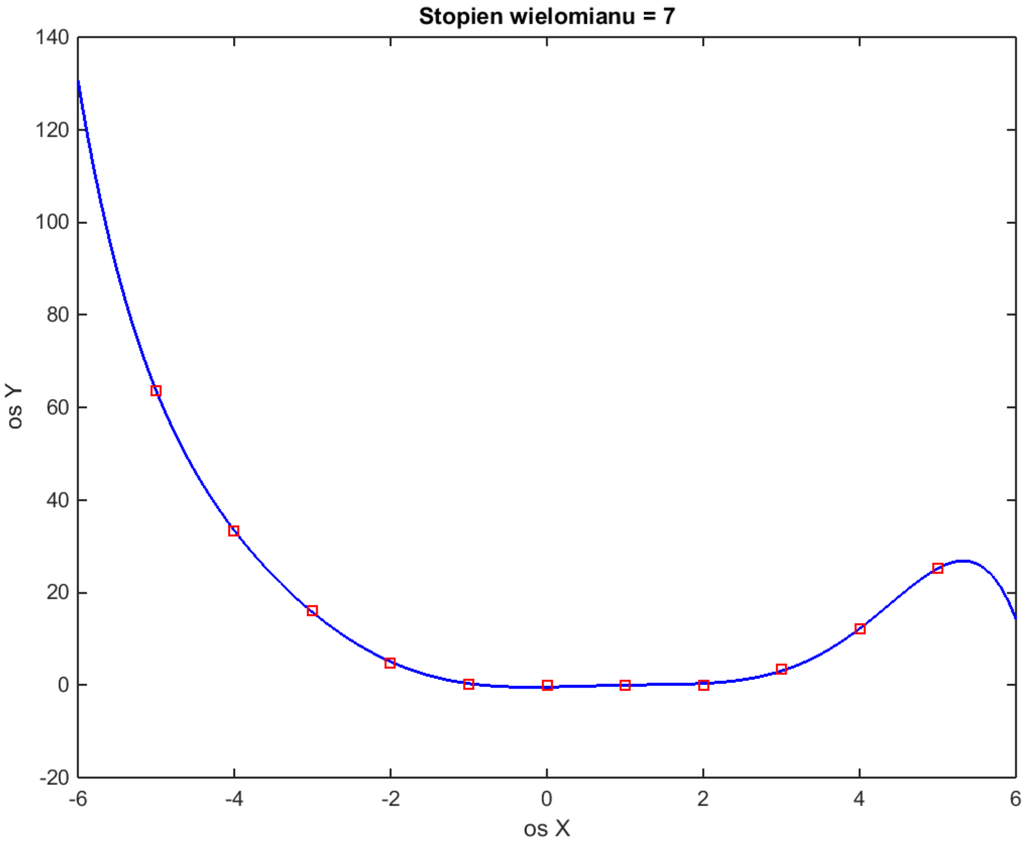


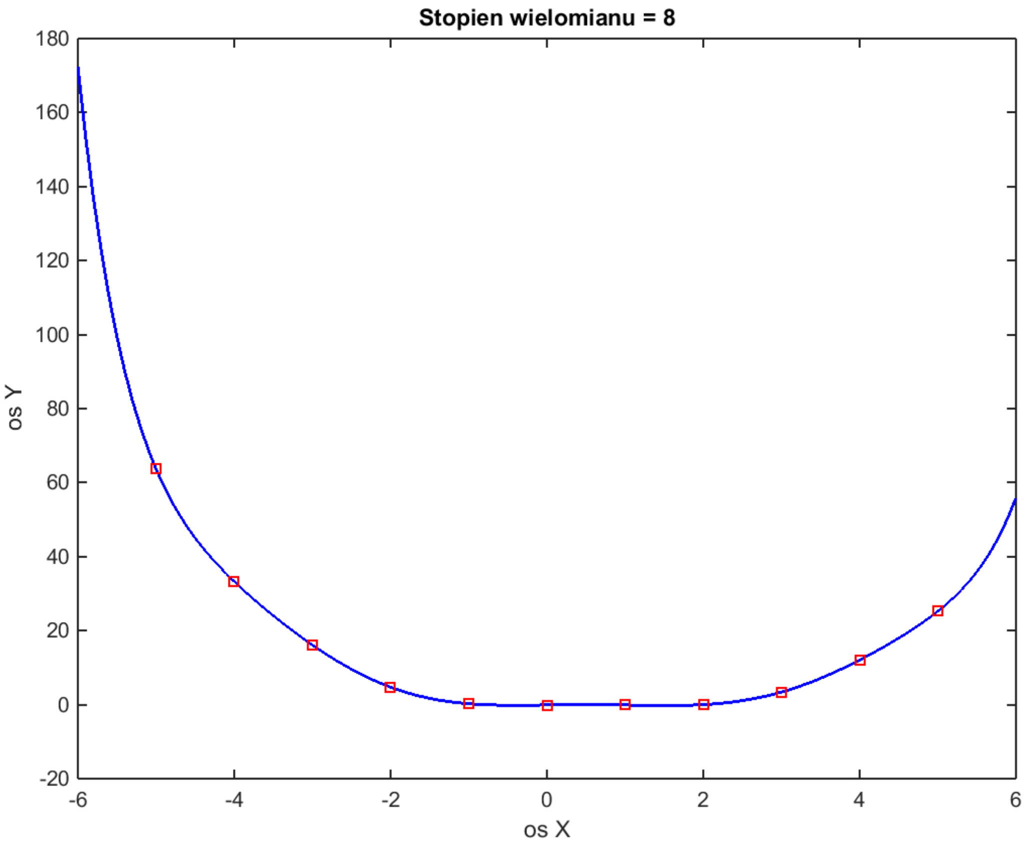

Stopien wielomianu = 3
100
80
60
40
20
0
-20
-6
-4
-2
0
2
4
6
os X
os Y

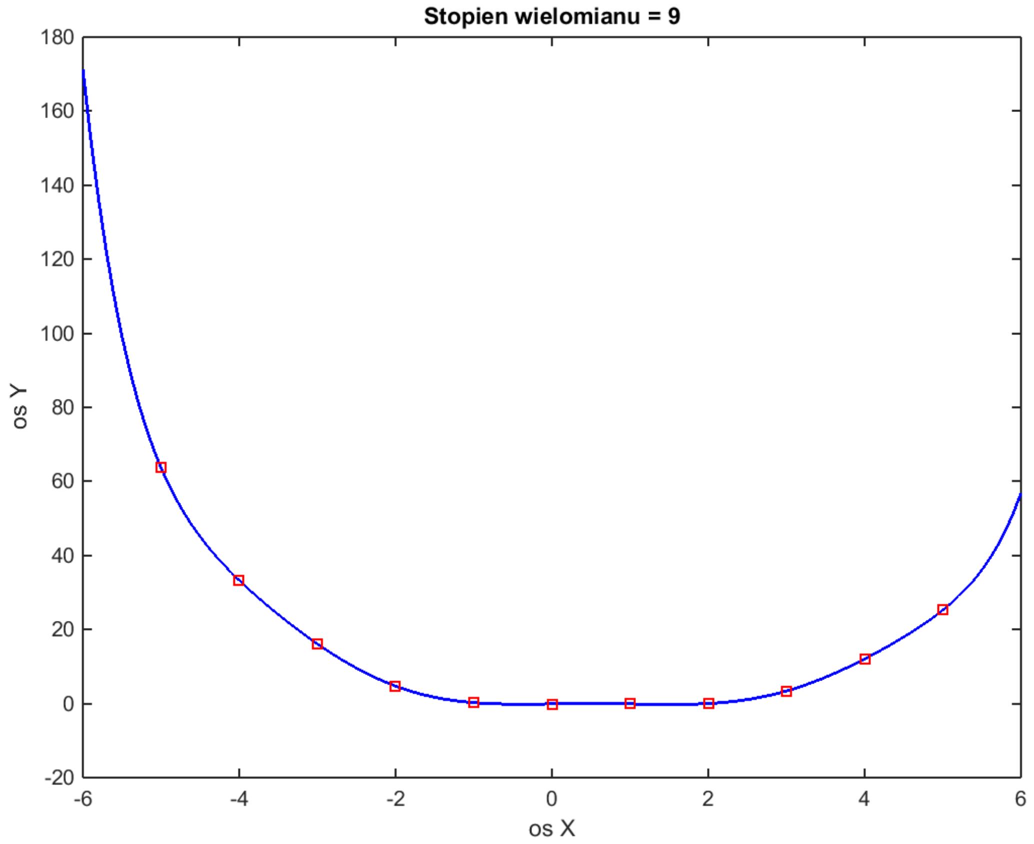

Stopien wielomianu = 4
120
100
80
60
40
20
0
-20
-6
-4
-2
0
2
4
6
os X
os Y

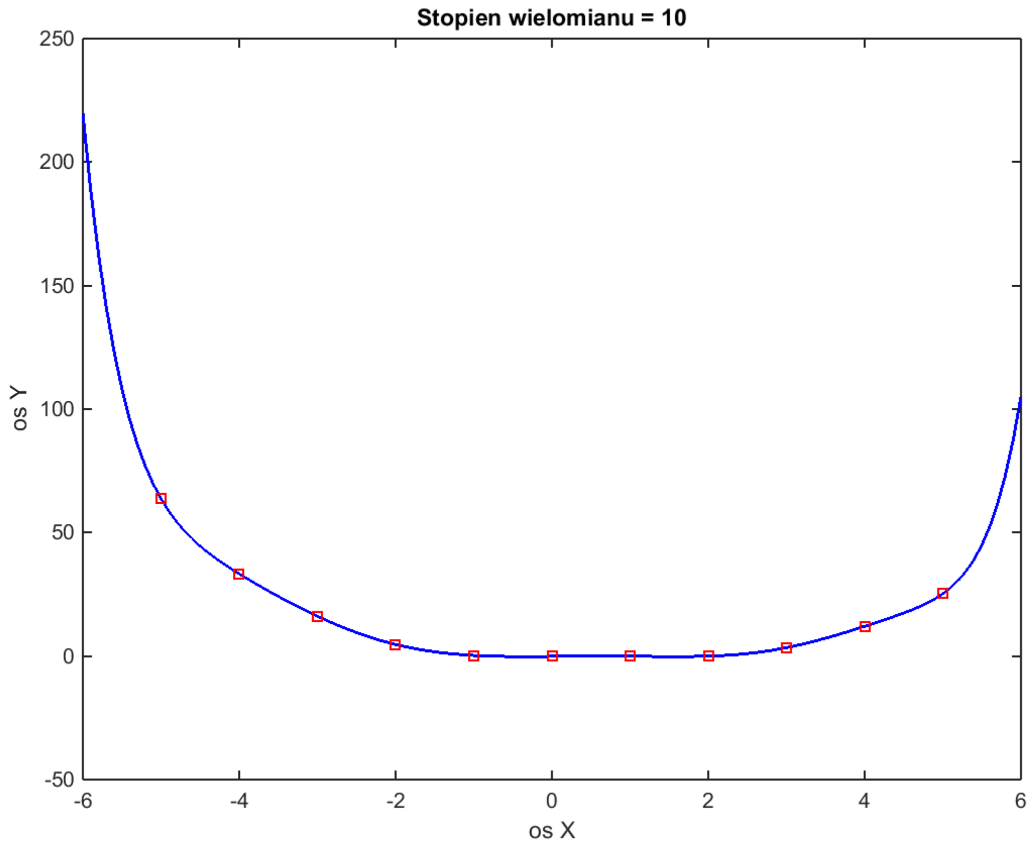

Stopien wielomianu = 5
120
100
80
60
40
20
0
-20
-6
-4
-2
0
2
4
6
os X
os Y


Stopien wielomianu = 6
120
100
80
60
40
20
0
-20
-6
-4
-2
0
2
4
6
os X
os Y


Stopien wielomianu = 7
140
120
100
80
60
40
20
0
-20
-6
-4
-2
0
2
4
6
os Y
os X


Stopien wielomianu = 8
180
160
140
120
100
80
60
40
20
0
-20
-6
-4
-2
0
2
4
6
os Y
os X


Stopien wielomianu = 9
180
160
140
120
100
80
60
40
20
0
-20
os Y
-6
-4
-2
0
2
4
6
os X


Stopien wielomianu = 10
250
200
150
100
50
0
-50
os Y
-6
-4
-2
0
2
4
6
os X

Komentarz do otrzymanych wyników oraz wnioski z eksperymentów (ocena poprawności wyników, dokładności, efektywności algorytmów itd.):

Można przede wszystkim zauważyć, że oba sposoby różnią się czasami wykonania. Czas wykonania dla układów równań normalnych jest znacznie mniejszy od czasu wykonania dla układów równań liniowych z macierzą R wynikającą z rozkładu QR macierzy układu równań problemu. Wynika to z faktu, iż w układach równań liniowych z macierzą R oprócz wykonywania analogicznych operacji co w układach równań normalnych, wykonywana jest dodatkowo operacja rozkładu QR macierzy. Powoduje to znaczny wzrost czasu wykonania. Dla pierwszych wartości stopni wielomianu błędy rozwiązania są w tym przypadku identyczne. Dopiero dla stopnia wielomianu równego 10 błędy rozwiązania zaczynają się różnić. Jeśli chodzi o wskaźnik uwarunkowania macierzy R to jest on znacznie mniejszy od wskaźnika uwarunkowania macierzy A^T*A. Możemy się o tym przekonać wykorzystując funkcje zaimplementowaną w Matlab: cond(). Jednak w przypadku wielomianów początkowych stopni nie wpływa to specjalnie na obliczenia, błędy rozwiązań są takie same. Wraz ze zwiększaniem stopnia wielomianu, zaczynają pojawiać się błędy numeryczne. Wartości macierzy A zaczynają być zdecydowanie większe od wartości macierzy X oraz Y podanych w treści zadania. Dlatego dla większych stopni wielomianu aproksymacja ta staje się coraz gorsza, błędy numeryczne są coraz większe. Zarówno dla układów równań normalnych jak i dla układów równań liniowych z macierzą R, w przypadku mniejszych stopni wielomianu, wraz ze wzrostem stopnia wielomianu aproksymującego błąd rozwiązania maleje. Jest to oczywista rzecz, ponieważ czym większy stopień wielomianu, tym bardzie „plastyczna” staje się funkcja i można tak dobrać współczynniki, że aproksymacja staje się dokładniejsza (przy czym elementy macierzy A pozostają względnie nieduże). O poprawności implementacji algorytmów można przekonać się analizując rysunki z funkcjami oraz danymi. Widać na nich, jak kolejne wielomiany coraz bardziej dopasowują się do podanych danych. Na podstawie wykresów otrzymanych w tym zadaniu można wywnioskować, iż zaimplementowane algorytmy poprawnie rozwiązują liniowe zadanie najmniejszych kwadratów.