

BADANIE PRZYDATNOŚCI PROCEDUR ROSENSTEINA I ECKMANNA DO IDENTYFIKACJI CHAOTYCZNYCH SZEREGÓW CZASOWYCH

USEFULNESS STUDY OF ROSENSTEIN AND ECKMANN PROCEDURES FOR IDENTIFICATION OF CHAOTIC TIME SERIES

Damian Hallmann, Piotr Jankowski*

Akademia Morska w Gdyni, Morska 81-87, 81-225 Gdynia, Wydział Elektryczny,
Katedra Elektroenergetyki Okrętowej, e-mail: p.jankowski@we.am.gdynia.pl

* Adres do korespondencji/Corresponding author

Streszczenie: Artykuł przedstawia wyniki badań symulacyjnych z użyciem procedur Eckmanna i Rosensteina wyznaczających wykładniki Lapunowa na podstawie szeregu czasowego. Dla weryfikacji i oceny przydatności tych procedur, jako wzorcowy szereg czasowy wykorzystano punkty generowane przez odwzorowanie logistyczne, dla którego znana jest trajektoria współczynników Lapunowa.

Słowa kluczowe: chaos, szereg czasowy, współczynnik Lapunowa.

Abstract: This paper presents the results of simulation tests using the Eckmann and Rosenstein procedures for calculating Lyapunov exponents based on a time series. For verifying and evaluating the suitability of these procedures as a reference time series, points generated by logistic mapping for which the trajectory of Lyapunov's coefficients is known was applied.

Keywords: chaos, time series, Lyapunov exponent.

1. WSTĘP

Teoria chaosu staje się coraz częściej narzędziem wykorzystywanym w badaniu nieliniowych układów dynamicznych. Nieregularną i nieprzewidywalną ewolucję takich systemów nieliniowych zaobserwowano już dawno zarówno w układach teoretycznych, czyli w rozwiązaniach modeli matematycznych, jak i podczas obserwacji (pomiarów) rzeczywistych obiektów fizycznych. Na dodatek coraz częściej badane obiekty stanowią bardzo złożone układy, takie jak np. systemy energetyczne. Dla takich układów jednoznaczne sformułowanie równań stanu, opisujących zachowanie tego systemu, jest na ogół bardzo trudne, a czasem niemo-

żliwe, gdyż model musiałby uwzględniać nie tylko parametry wszystkich obiektów, z których składa się system, ale także stany awaryjne oraz zakłócenia natury zarówno deterministycznej, jak i losowej. W takiej sytuacji badanie zachowań chaotycznych sprowadza się do badań pomiarowych charakterystyk czasowych, dla których nie jest znana a priori wymiarowość przestrzeni stanu. Wówczas można próbować dokonać rekonstrukcji tej przestrzeni metodą opóźnień i wyznaczyć takie parametry jak wymiar przestrzeni fazowej (wymiar zanurzenia) i czas opóźnienia [Packard i in. 1980]. Parametry te są potrzebne do określenia jednego z podstawowych wskaźników identyfikującego zachowanie chaotyczne szeregu czasowego, jakim jest dodatnia wartość wykładnika Lapunowa.

Należy podkreślić, iż wielu autorów uważa, że nie istnieje jednoznaczna metoda stosowalna dla każdego zbioru danych, wyznaczająca minimalny wymiar zanurzenia i czas opóźnienia. W pracach tych zaleca się wręcz subiektywny sposób postępowania polegający na metodzie prób [Kantz 1994; Peters 1997].

W prezentowanym artykule autorzy przeprowadzili badania przydatności dwóch programów wyznaczających wykładniki Lapunowa na podstawie procedur Eckmanna i Rosensteina. Obie procedury bazują na metodzie opóźnień i związanej z nią metodzie fałszywych sąsiadów [Packard i in. 1980], przy czym procedura Eckmanna i innych jako argument przyjmuje macierz danych, czyli wyznaczone szeregi dla więcej niż jednej zmiennej stanu. Argumentem procedury Rosensteina i innych jest tylko jeden szereg i stąd wyznacza ona wyłącznie największy wykładnik Lapunowa.

Badanie przydatności powyższych procedur przeprowadzono poprzez porównanie z trajektorią wzorcową, za którą przyjęto charakterystykę wykładników Lapunowa dla odwzorowania logistycznego. Trajektorię wzorcową można wyznaczyć wprost z definicji dzięki znanej postaci analitycznej tego odwzorowania. Przy zmieniającym się współczynniku a w zakresie $\langle 2,5;4 \rangle$ w pewnych przedziałach, generowane szeregi charakteryzują się stanem stabilnym, a w innych przechodzą w stan chaotyczny.

2. WYKŁADNIK LAPUNOWA

Założono, że istnieje pewne przekształcenie generujące ciąg wyników w danej postaci: $x_{k+1} = f(x_k)$ oraz para punktów startowych $(x_0, x_0 + \Delta_0)$, różniących się o niewielką wartość błędu Δ_0 . Jako przekształcenie wybrano liniowe lub założono, że odległości pomiędzy dwoma punktami są tak niewielkie, że przekształcenie $f(x_k)$ można aproksymować funkcją liniową $x_{k+1} = cx_k$. Trzeba obliczyć kilka punktów trajektorii, które będą powstawać, startując z dwóch różnych punktów (1) (pomijając fakt, że w rzeczywistości współczynniki kierunkowe c w każdym kroku ulegają zmianie):

$$\begin{aligned} \widehat{x}_1 &= f(x_0 + \Delta_0) = c(x_0 + \Delta_0) & x_1 &= f(x_0) = cx_0 \Rightarrow \widehat{x}_1 - x_1 = \Delta_0 \\ \widehat{x}_2 &= f(\widehat{x}_1) = c^2(x_0 + \Delta_0) & x_2 &= f(x_1) = c^2x_0 \Rightarrow \widehat{x}_2 - x_2 = \Delta_1 = c^2\Delta_0 \end{aligned} \quad (1)$$

.....

$$\widehat{x}_n = f(\widehat{x}_{n-1}) = c^n(x_0 + \Delta_0) \quad x_n = f(x_1) = c^n x_0 \Rightarrow \widehat{x}_n - x_n = \Delta_n = c^n \Delta_0$$

Należy zauważyć, że:

$$c^n = \frac{\Delta_n}{\Delta_0} = \frac{\Delta_n}{\Delta_{n-1}} \cdot \frac{\Delta_{n-1}}{\Delta_{n-2}} \cdot \dots \cdot \frac{\Delta_1}{\Delta_0} \quad (2)$$

Po zlogarytmowaniu (2) otrzymuje się:

$$\ln|c| = \frac{1}{n} \ln \left| \frac{\Delta_n}{\Delta_0} \right| = \frac{1}{n} \sum_{k=1}^n \ln \left| \frac{\Delta_k}{\Delta_{k-1}} \right| \quad (3)$$

Po rozpisaniu Δ_k, Δ_{k-1} (4)

$$\left. \begin{aligned} \Delta_{k-1} &= \widehat{x}_{k-1} - x_{k-1} \Rightarrow \widehat{x}_{k-1} = x_{k-1} + \Delta_{k-1} \\ \Delta_k &= \widehat{x}_k - x_k = f(\widehat{x}_{k-1}) - f(x_{k-1}) \end{aligned} \right\} \Rightarrow \Delta_k = f(x_{k-1} + \Delta_{k-1}) - f(x_{k-1}) \quad (4)$$

Mamy:

$$\frac{\Delta_k}{\Delta_{k-1}} = \frac{f(x_{k-1} + \Delta_{k-1}) - f(x_{k-1})}{\Delta_{k-1}} \Rightarrow \lim_{\Delta_{k-1} \rightarrow 0} \frac{\Delta_k}{\Delta_{k-1}} = f'(x_{k-1})$$

Wówczas w granicy do nieskończoności, prawą stronę wyrażenia (3) można zapisać:

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=1}^n \ln|f'(x_{k-1})| = \lambda(x_0) \quad (5)$$

Współczynnik $\lambda(x_0)$ jest określany mianem wykładnika Lapunowa. Wzór (5) można wyprowadzić wprost z definicji wykładnika Lapunowa, określanego jako miara oddalania się punktów, $(x_0, x_0 + \Delta_0)$ równa po n iteracjach liczbie:

$$\Delta_0 e^{n\lambda(x_0)} = f^n(x_0 + \Delta_0) - f^n(x_0) \quad (6)$$

Wówczas przechodząc do granicy ($\Delta_0 \rightarrow 0, n \rightarrow \infty$), uzyskuje się wzór na $\lambda(x_0)$:

$$\lim_{n \rightarrow \infty} \lim_{\Delta_0 \rightarrow 0} \frac{1}{n} \ln \left| \frac{f^n(x_0 + \Delta_0) - f^n(x_0)}{\Delta_0} \right| = \lim_{n \rightarrow \infty} \frac{1}{n} \ln \left| \frac{df^n(x_0)}{dx_0} \right|$$

Z kolei, wykorzystując regułę różniczkowania funkcji złożonej, ostatecznie otrzymuje się wzór (7) identyczny z wyżej określoną formułą (5):

$$\lim_{n \rightarrow \infty} \frac{1}{n} \ln \left| \frac{df^n(x_0)}{dx_0} \right| = \lim_{n \rightarrow \infty} \frac{1}{n} \ln \left| \prod_{k=1}^n f'(x_{k-1}) \right| = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=1}^n \ln |f'(x_{k-1})| = \lambda(x_0) \quad (7)$$

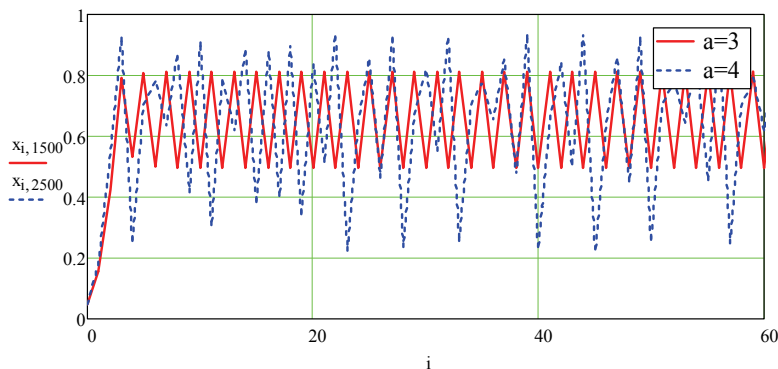
3. PRZEKSZTAŁCENIE LOGISTYCZNE

Przekształcenie logistyczne (8) jest najbardziej chyba znanym odwzorowaniem wśród badaczy teorii chaosu [Peitgen, Jurgens i Saupe 1996; Baker i Gollub 1998], podawanym jako przykład zmiany zachowania trajektorii ze stabilnej do chaotycznej w zależności od współczynnika a :

$$x_{i+1} = ax_i(1 - x_i) \quad (8)$$

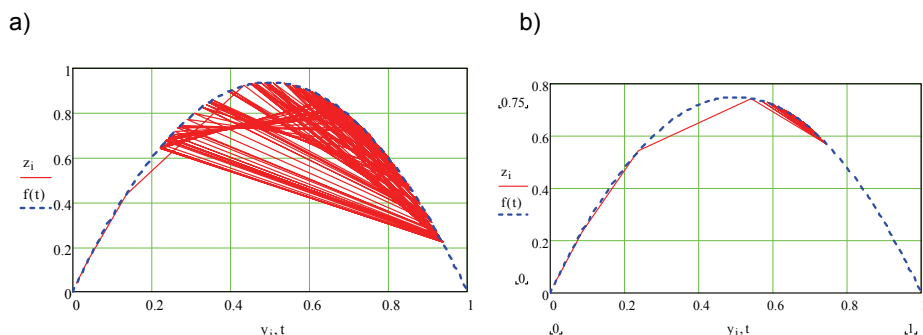
Na rysunku 1 przedstawiono dwa przebiegi wygenerowane z rekurencyjnego wzoru (8) w środowisku Mathcad dla dwóch parametrów $a = 3$ oraz $a = 4$. Z rysunku 1 wyraźnie wynika, że uzyskana trajektoria dla pierwszego parametru ma charakter stabilizujący się na poziomie dwóch wartości 0,495 oraz 0,812, natomiast dla parametru $a = 4$ przebieg zachowuje się chaotycznie, przyjmując niepowtarzalne wartości w kolejnych iteracjach. Wyjaśnienie takiej zmiany w charakterze przebiegów przedstawia sposób iterowania poszczególnych punktów odwzorowania logistycznego na podstawie wykresu paraboli (rys. 2).

W celu przedstawienia globalnego zachowania się modelu powszechnie (nie tylko dla odwzorowania logistycznego) wyznacza się diagram bifurkacyjny, w którym współczynnik a (lub μ) zmienia się w wybranym przedziale w sposób ciągły (w symulacji, co pewien zadany krok). Dla wybranej wartości początkowej przeprowadza się np. N -iteracji (najczęściej opuszczając 100 pierwszych) i na diagramie dla danego μ zaznacza się pozostałe iteracje. Znając postać analityczną przekształcenia logistycznego, można również wyznaczyć przebieg zmienności wykładnika Lapunowa, stosując bezpośrednio wzór (5).



Rys. 1. Przebieg trajektorii odwzorowania logistycznego stabilnej i chaotycznej

Fig. 1. Trajectories of logistic map (stable and chaotic state)



Rys. 2. Sposób iterowania punktów przekształcenia logistycznego: a) dla $a = 3$, b) $a = 4$

Fig. 2. The way to iterate logistic transformation points for: a) $a = 3$, b) $a = 4$

Procedurę wyznaczania diagramu bifurkacyjnego, w której współczynnik odwzorowania oznaczono przez μ , i jednocześnie wykładnika Lapunowa dla zmieniającego się tutaj współczynnika a przedstawiono poniżej.

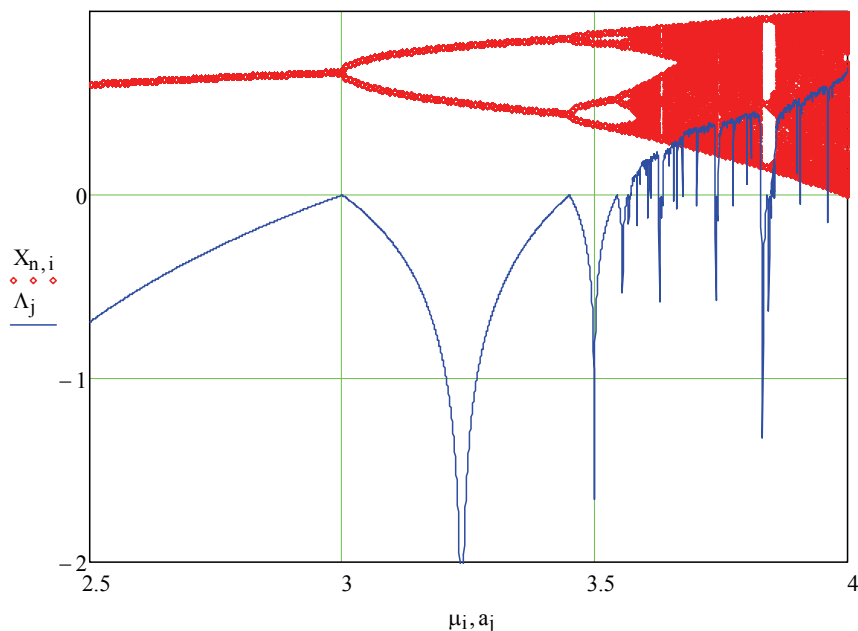
Procedura bifurkacji dla odwzorowania logistycznego:

$$\begin{aligned} N_i &:= 500 & i &:= 0..N & n &:= 0..N & \mu_i &:= 4 \cdot \frac{i}{N} \\ X_{0,i} &:= 0.01 & X_{n+1,i} &:= \mu_i \cdot X_{n,i} \cdot (1 - X_{n,i}) & n &:= \frac{N}{5} .. N \end{aligned}$$

Procedura wyznaczania wykładnika Lapunowa dla odwzorowania logistycznego:

$$M := 2000 \quad j := 0..3000 \quad a_j := 2.5 + j \cdot 0.0005 \quad k := 0..M \quad x_{0,j} := 0.01$$

$$x_{k+1,j} := a_j \cdot x_{k,j} \cdot (1 - x_{k,j}) \quad \Lambda_j := \frac{1}{M-100} \cdot \sum_{k=101}^M \ln[|a_j \cdot (1 - 2 \cdot x_{k,j})|]$$



Rys. 3. Diagram bifurkacyjny i wykładnik Lapunowa

Fig. 3. Bifurcation diagram and Lyapunov exponent

Powyzsza trajektoria Lapunowa (rys. 3) wyznaczona dla zmieniajacej sie wartosci wspolczynnika μ (lub a) i uzyskana na podstawie analitycznego wzoru (8) bedzie stanowila wzorcowa trajektorie przy badaniach procedur (Eckmanna i Rosensteina), stosowanych do wyznaczania wykladnikow Lapunowa na podstawie szeregow danych pomiarowych.

4. PROCEDURY ECKMANNA I ROSENSTEINA

W literaturze teorii chaosu deterministycznego istnieje wiele prac opisujacych procedury wyznaczajace spectrum wykladnikow Lapunowa dla eksperymentalnych szeregow czasowych, a nawet udostepniajacych ich implementacje w roznych jezykach (QBasic, Fortran, Python, Matlab) [Wolf i in. 1985; Eckmann i in. 1986; Rosenstein, Collins i De Luca 1993]. Najczesciej powyzsze procedury opieraja sie na wyznaczeniu parametrów (rekonstrukcji) przestrzeni fazowej oraz metodzie falszywych najblizszych sasiedow (patrz poprzedni artykul autorstwa P. Jankowskiego). Poniewaz jak wspomniano, w spektrum Lapunowa najbardziej interesujacym wskaźnikiem stanu chaosu (badz jego braku) jest wartosc maksymalnego wykladnika Lapunowa, to do badan przydatnosci takich procedur wybrano oprócz procedury Eckmanna i innych dodatkowo algorytm Rosensteina i innych liczacej

jeden maksymalny wykładnik [Eckmann i in. 1986; Rosenstein, Collins i De Luca 1993]. Obie procedury zostały zrealizowane w programie Python [*Largest Lyapunov Exponent with Rosenstein's Algorithm*].

W opisie algorytmu Rosensteina znamienne jest pierwsze zdanie: „Algorytm Rosensteina i in. jest w stanie odzyskać największy wykładnik Lapunowa, ale jego wartość silnie zależy od wyboru parametrów”.

Moduł realizujący algorytm Rosensteina w programie Python zawiera następujące argumenty i parametry:

nolds.lyap_r(*data*, *emb_dim*=10, *lag*=None, *min_step*=None, *tau*=1, *min_vectors*=20, *trajectory_len*=20, *fit*='RANSAC', *debug_plot*=False, *plot_file*=None)

Parametrami w module, obliczającymi ten wykładnik, są:

- *data* – jednowymiarowy szereg czasowy;
- *emb_dim* – wymiar zanurzenia (wymiar zrekonstruowanej przestrzeni);
- *lag* – opóźnienie czasowe w metodzie opóźnień rekonstrukcji przestrzeni fazowej;
- *min_tstep* – minimalny odstęp czasowy pomiędzy dwoma sąsiadami (domyślnie program znajduje odpowiednią wartość, obliczając średni przedział czasowy danych);
- *tau* – rozmiar kroku pomiędzy punktami (danymi) w szeregu czasowym w sekundach (domyślnie znajduje odpowiednią wartość za pomocą funkcji autokorelacji);
- *min_vectors* – jeśli *lag* jest ustawione jako *None*, szukanie odpowiedniego opóźnienia zostanie zatrzymane, gdy liczba powstałych wektorów spadnie poniżej *min_vectors*;
- *trajectory_len* – czas (w liczbie punktów danych), śledzenia zmiany odległości między dwoma sąsiednimi punktami;
- *fit* – jeżeli *fit* jest ustawione na "poli", metodą aproksymacji jest metoda najmniejszych kwadratów, jeżeli na 'RANSAC' – stosowana jest procedura 'RANSAC', która jest bardziej odporna na odchylenia.

Pozostałe parametry, takie jak *debug_plot* oraz *plot_file*, dotyczą opcji pojawienia się (lub nie) oszacowań pośrednich oraz wykresu.

Moduł Eckmanna odwołuje się ogólnie do tej samej metody (najbliższych sąsiadów), ale podstawowym argumentem może być, nie jak w przypadku procedury Rosensteina jeden szereg, lecz tablica szeregów, co oznacza, że program ten może zwrócić spektrum wykładników Lapunowa (WL), odpowiadających tym szeregom. Jeżeli daną wejściową jest jeden szereg, to program generuje największy wykładnik (nawet jeżeli z rekonstrukcji przestrzeni wynika, że jej wymiar jest większy niż jeden). W module Eckmanna rezultat widma Lapunowa zależny jest od następujących argumentów i parametrów:

nolds.lyap_e(*data*, *emb_dim*=10, *matrix_dim*=4, *min_nb*=None, *min_tsep*=0, *tau*=1, *debug_plot*=False, *debug_data*=False, *plot_file*=None).

Opis parametrów jest następujący:

- *data* – szereg skalarny lub tablica danych;
- *emb_dim* – wymiar zanurzenia;
- *matrix_dim* – wymiar macierzy wektorów historii (*emb_dim*-1 musi być podzielne przez *matrix_dim*-1);
- *min_nb* – minimalna liczba sąsiadów (domyślna wartość = $\min(2*matrix_dim, matrix_dim + 4)$);
- *tau* – rozmiar kroku danych w sekundach (współczynnik skalowania (normalizacji) dla wykładników).

Chociaż podobnie jak dla procedury Rosensteina moduł Eckmanna nie wyznacza automatycznie wszystkich parametrów, jednak w instrukcji użytkownika pomocne są następujące rekomendacje:

- wzrost liczby danych poprawia dokładność, mały *tau* nie;
- użyć dużych wartości dla *emb_dim*;
- wymiar *matrix_dim* powinien być "nieco większy niż przewidywana liczba dodatnich wykładników Lapunowa";
- zaleca się ustawić parametr $min_nb = \min(2*matrix_dim, matrix_dim + 4)$.

Do badania przydatności powyżej przedstawionych procedur przyjęto za niezmiennie pewne parametry, zwłaszcza te, które w module są już ustawione na zalecaną wartość oraz te ze wskaźnikiem *None*, co oznacza, że moduł w trakcie rekonstrukcji wsadowego szeregu określi je samodzielnie.

Wartości najważniejszych parametrów, dla których przeprowadzono badania, przedstawiono dla procedury Eckmanna w tabeli 1, a dla procedury Rosensteina – w tabeli 2.

Tabela 1. Ustawienie parametrów dla procedury Eckmanna

Table 1. Setting parameters for the Eckmann procedure

Liczba punktów	<i>emb_dim</i>	<i>matrix_dim</i>	<i>min_nb</i>	<i>min_tstep</i>	<i>tau</i>
300	3,5,7,9	3	None	0	1

Tabela 2. Ustawienie parametrów dla procedury Rosensteina

Table 2. Setting parameters for the Rosenstein procedure

Liczba punktów	<i>emb_dim</i>	<i>lag</i>	<i>min_tstep</i>	<i>tau</i>	<i>min_vectors</i>	<i>trajectory_len</i>
300	3,5,7,9	None	None	1	20,30,40,50	20,30,40,50

5. ANALIZA WYNIKÓW

Procedura Eckmanna zapewnia (oprócz wymiaru zanurzenia (*emb_dim*) i wymiaru macierzy (*matrix_dim*)) automatyczne wyznaczenie pozostałych parametrów przy jednoczesnym ustawieniu parametru *tau* na wartości 1. Ustawienie to jest oczy-

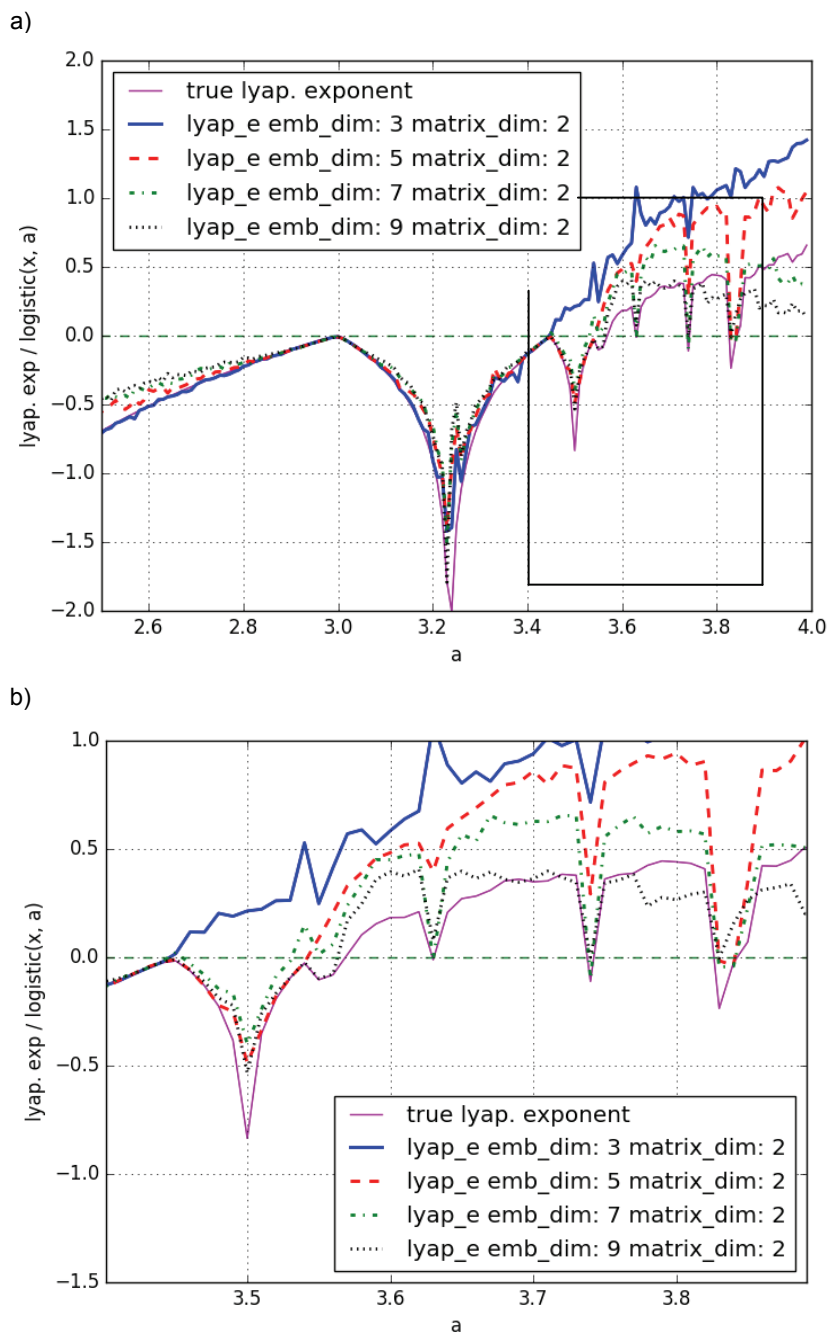
wiste z powodu iteracyjnego sposobu generowania kolejnych punktów odwzorowania logistycznego. W związku z tym badania przydatności tej procedury przeprowadzono, wyznaczając rodzinę przebiegów wykładnika Lapunowa w funkcji współczynnika a dla różnych parametrów (emb_dim) oraz ($matrix_dim$). Jak wspomniano wcześniej, musi zachodzić warunek $(matrix_dim-1) \mid (emb_dim-1)$, stąd wartość parametru $matrix_dim = 2$ umożliwia ustawienie dowolnych wartości parametru emb_dim .

Na rysunku 4 przedstawiono pierwszą rodzinę charakterystyk wykładników Lapunowa (WL) przy zmieniającym się parametrze emb_dim i stałym $matrix_dim = 2$ na tle wzorcowej charakterystyki analitycznej odwzorowania logistycznego. Analiza tych przebiegów wskazuje duży wpływ wymiaru zanurzenia (emb_dim) na wyznaczone wartości WL, przy czym wpływ ten jest większy dla wartości współczynnika a powyżej 3,5, tj. dla zakresu, gdzie wykładniki przybierają wartości dodatnie.

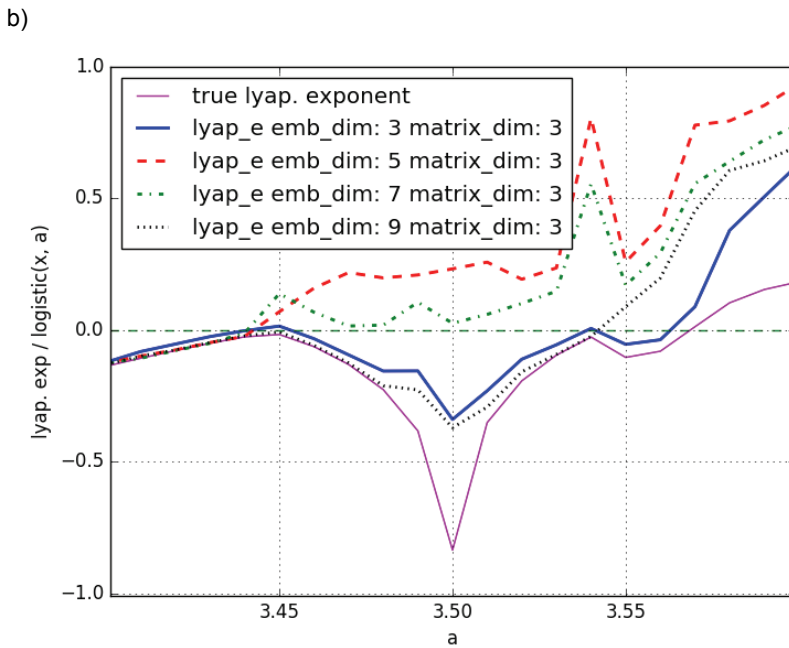
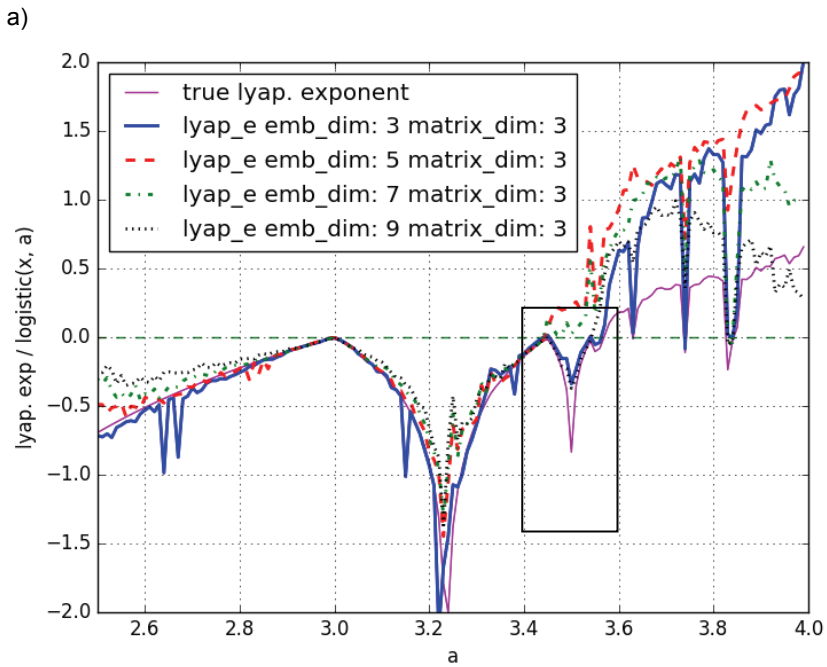
Na rysunku 4b pokazano powiększone okno z rysunku 4a w zakresie $3,4 < a < 4$. Tu widać, że najbliższej trajektorii wzorcowej jest przebieg dla $emb_dim = 9$. Jednocześnie, co jest bardzo istotne, ten przebieg prawie zawsze zachowuje znak wykładnika taki jak przebiegu wzorcowego.

Na rysunku 5 przeprowadzono symulacje dla tych samych parametrów wymiaru zanurzenia przy ustawieniu $matrix_dim = 3$. Dla tych parametrów również zachowany jest warunek $(matrix_dim-1) \mid (emb_dim-1)$. Obserwując przebiegi, można zauważyć, że nastąpiła większa rozbieżność niż w poprzednim przypadku zwłaszcza dla $a > 3,4$. Co gorsza, obserwując okno dla zakresu $3,4 < a < 3,6$ (rys. 5b), przebieg o wymiarze $emb_dim = 9$ nie zachowuje już znaku wykładnika takiego jak przebiegu wzorcowy.

W przypadku badań przydatności procedury Rosensteina analizowany był wpływ zmian jednego z trzech parametrów (tab. 2) na trajektorie Lapunowa przy niezmienności pozostałych. Podobnie jak w badaniach wcześniejszych (Eckmann) parametry, które nie wymagają podania konkretnej wartości (np. ustawienie *None*), co znaczy, że procedura oblicza je samodzielnie, nie były poddane analizie zmienności. Na początek wyznaczono rodzinę trajektorii Lapunowa przy zmiennym wymiarze rekonstruowanej przestrzeni fazowej ($emb_dim = variab$) i pozostałych dwóch $min_vectors$ oraz $trajectory_len$, ustawionych na wartości odpowiednio 10 i 20 (rys. 6). Należy zauważyć, że tym razem wymiar zanurzenia nie wpłynął na przebieg trajektorii WL (wszystkie przebiegi pokrywają się), ale zgodność z przebiegiem teoretycznym jest gorsza w porównaniu z procedurą Eckmanna dla tego samego wymiaru zanurzenia (rys. 4). Ponadto, obserwując powiększone okno na rysunku 6b, widać kilka punktów, w których znak wykładnika Lapunowa różni się od znaku dla trajektorii teoretycznej. W tym zakresie sytuacja poprawiła się, jeżeli dla tej samej zmienności wymiaru zanurzenia powiększono $min_vectors$ do wartości 20 (rys. 7).



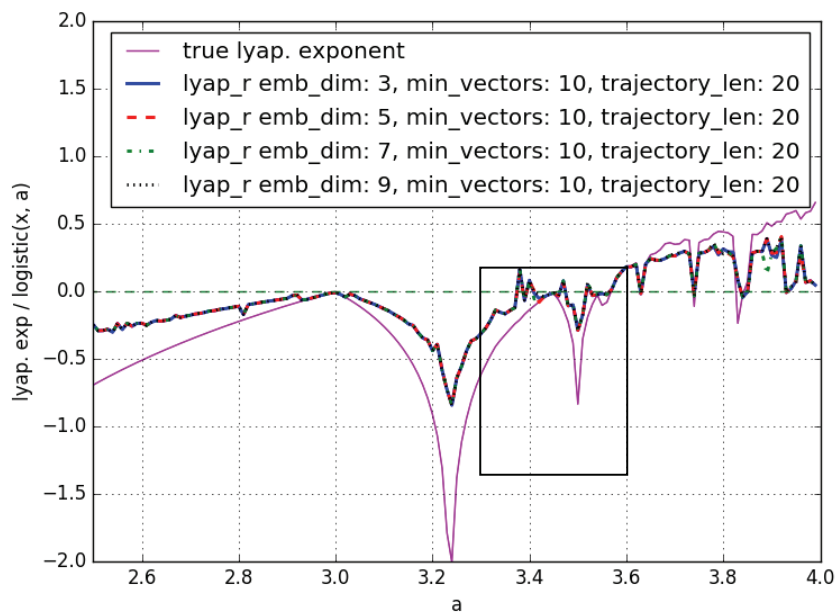
Rys. 4. Wykresy WL z procedury Eckmanna dla różnych wymiarów zanurzenia
Fig. 4. WL charts from Eckmann's procedure for different embedding dimensions



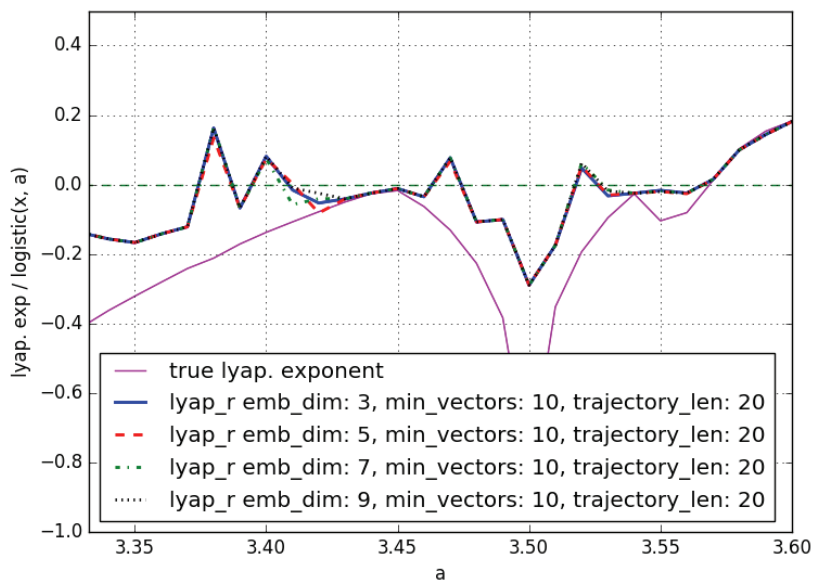
Rys. 5. Wykresy WL z procedury Eckmanna dla różnych wymiarów zanurzenia

Fig. 5. WL charts from Eckmann's procedure for different embedding dimensions

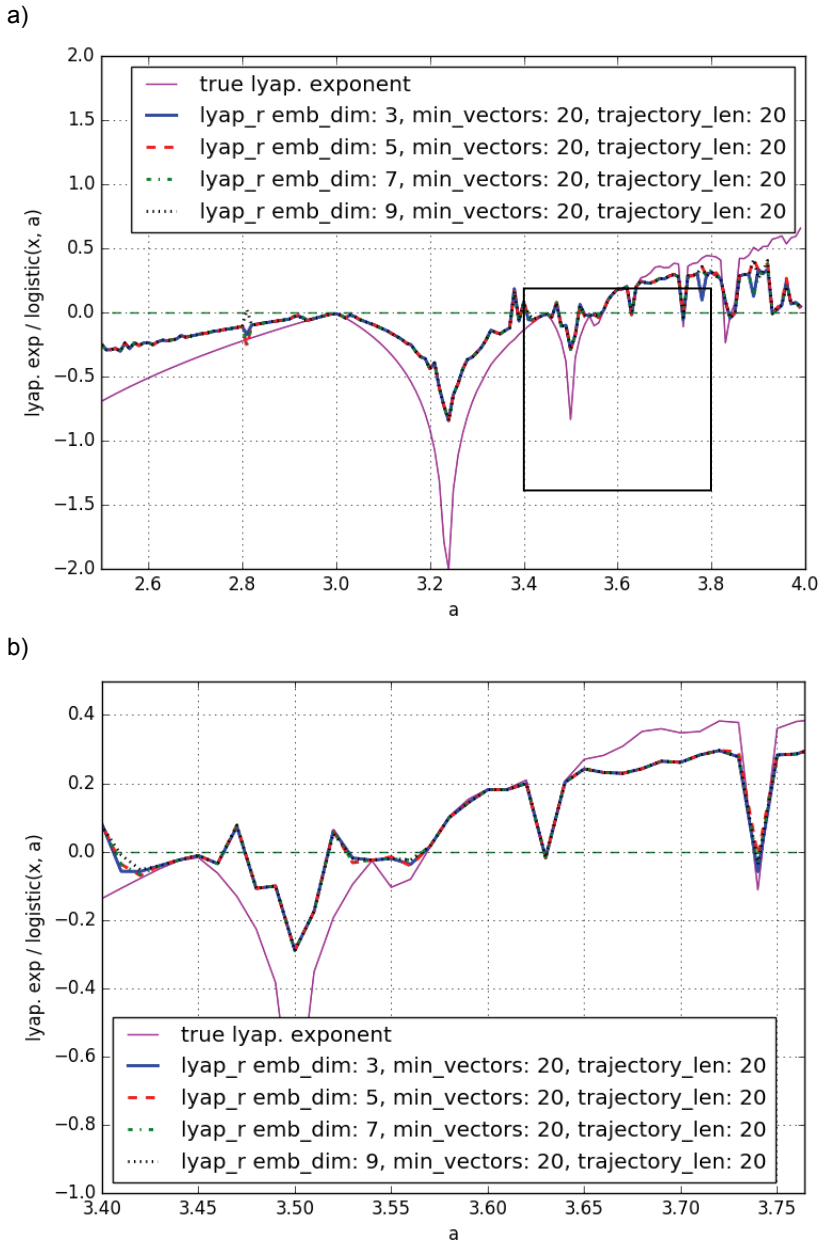
a)



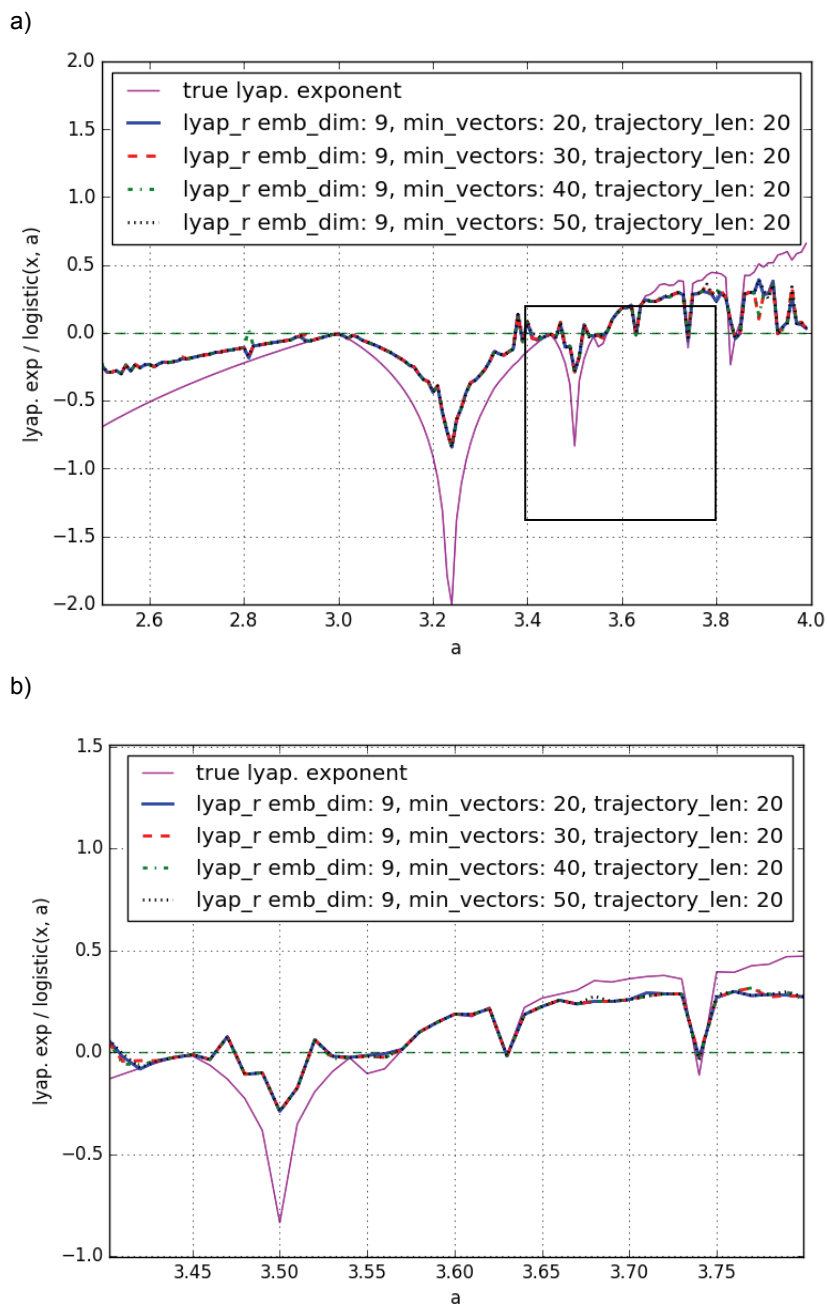
b)



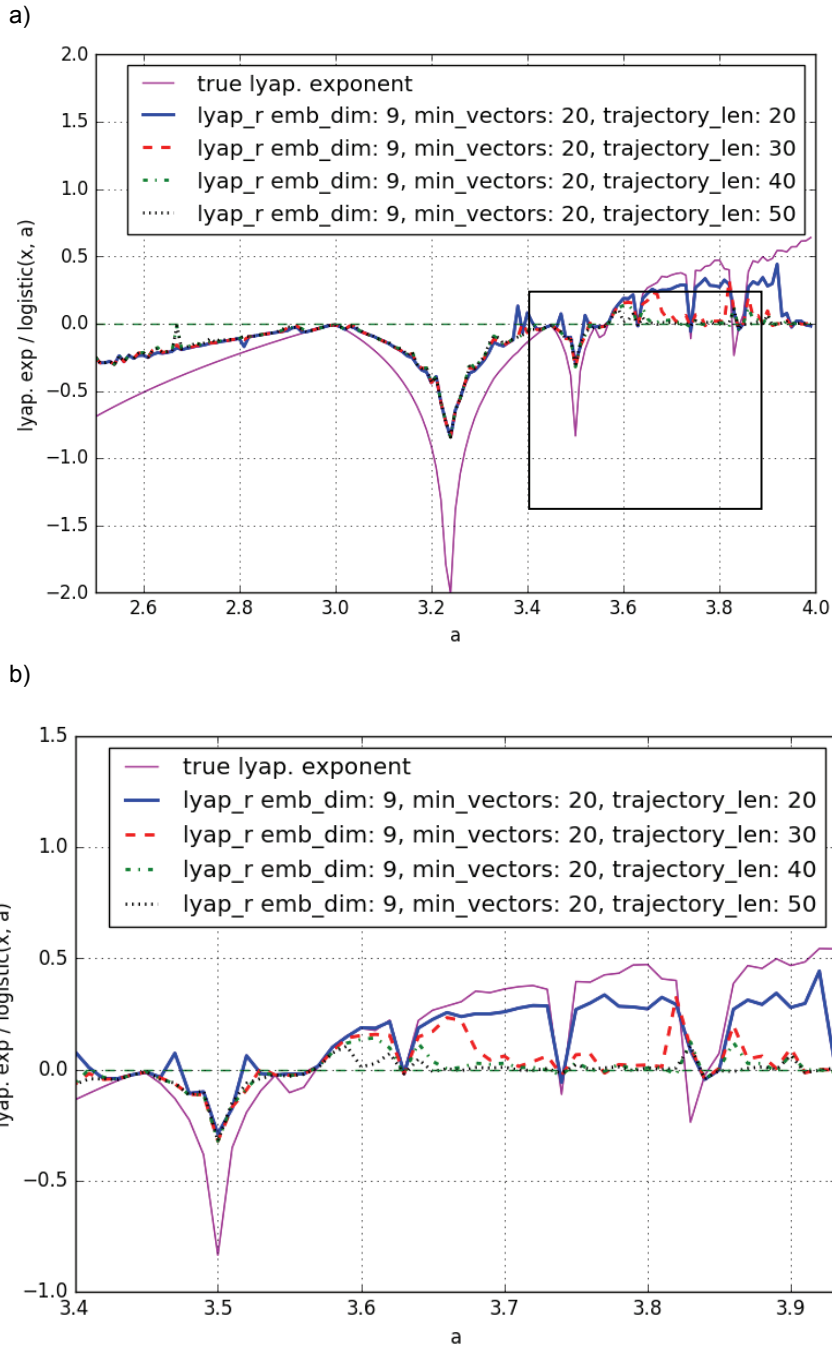
Rys. 6. Wykresy WL z procedury Rosensteina dla różnych wymiarów zanurzenia
Fig. 6. WL charts from Rosenstein's procedure for different embedding dimensions



Rys. 7. Wykresy WL z procedury Rosensteina dla różnych wymiarów zanurzenia
Fig. 7. WL charts from Rosenstein's procedure for different embedding dimensions



Rys. 8. Wykresy WL z procedury Rosensteina dla różnych parametrów $min_vectors$
Fig. 8. WL charts from Rosenstein's procedure for different $min_vectors$ parameters



Rys. 9. Wykresy WL z procedury Rosensteina dla różnych parametrów *trajectory_len*
Fig. 9. WL charts from Rosenstein's procedure for different *trajectory_len* parameters

Na podstawie wyników z rysunków 6 i 7 (i zalecenia w opisie procedury) w dalszych badaniach zmienności parametru *min_vectors* wymiar zanurzenia ustawiono na wartość 9 (rys. 8). Również i w tym przypadku nie odnotowano większego wpływu parametru *min_vectors* na przebieg trajektorii Lapunowa, natomiast w przedziale $3,4 < a < 3,8$ odnotowano nieznaczną poprawę zgodności z przebiegiem teoretycznym zarówno pod względem wartości, jak i znaku (rys. 8b). Na końcu zbadano wpływ zmienności parametru *trajectory_len* przy stałych parametrach *emb_dim* = 9 i *min_vectors* = 20 (rys. 9). Obserwacje wykresów w tym przypadku pokazują, że spośród wcześniej badanych parametrów największy wpływ na przebieg trajektorii WL ma parametr *trajectory_len*. Obserwując jednocześnie okno w zakresie $3,4 < a < 3,95$, można stwierdzić, że najbardziej zbliżona do wzorcowej jest trajektoria WL dla wartości parametru *trajectory_len* = 20.

6. PODSUMOWANIE

Celem przeprowadzonych badań symulacyjnych było sprawdzenie przydatności procedur Eckmanna i Rosensteina do wyznaczania wartości wykładników Lapunowa na podstawie jednowymiarowego szeregu danych. Porównując trajektorie uzyskane z tych procedur z przebiegiem wykładnika Lapunowa, określonego na podstawie wzoru (8) dla odwzorowania logistycznego, należy stwierdzić, że dla przeprowadzonych badań nie udało się uzyskać idealnej zgodności w całym zakresie zmian współczynnika odwzorowania (a), przy czym zgodność ta była lepsza w przedziale, dla którego wykładnik Lapunowa przyjmował wartości dodatnie. Natomiast bardziej pozytywny wniosek można postawić w przypadku zachowania znaków wykładników Lapunowa. Należy jednak podkreślić, że parametry, dla których wyznaczane były trajektorie Lapunowa (dla obu procedur), były stałe w całym zakresie zmieniającego się współczynnika a odwzorowania logistycznego. Z kolei każda zmiana tego współczynnika oznacza inny szereg generowany przez odwzorowanie i można sądzić, iż próba doboru parametrów dla każdej indywidualnej wartości a dałaby lepsze oszacowanie wykładnika Lapunowa.

Jak wynika z danych, umieszczonych w tabelach 1 i 2, symulacje były przeprowadzone dla 300 punktów. Wykonano kilka próbnych badań dla większej liczby punktów, ale nie odnotowano istotnych różnic pomimo ustawienia w obu algorytmach obciążenia pierwszych 10 punktów (większa liczba punktów obciążenia nie miała istotnego wpływu na wyniki symulacji).

Porównując obie procedury, można jednoznacznie stwierdzić, że bardziej stabilna i mniej czuła na zmianę parametrów zależnych od rekonstruowanej (nieznanej w praktyce) przestrzeni fazowej okazała się procedura Rosensteina.

LITERATURA

- Baker, G.L., Gollub, J.P., 1998, *Wstęp do dynamiki układów chaotycznych*, Wydawnictwo Naukowe PWN, Warszawa.
- Eckmann, J.P., Kamphorst, S.O., Ruelle, D., Ciliberto, S., 1986, *Liapunov Exponents from Time Series*, Physical Review A, vol. 34, no. 6, s. 4971–4979.
- Kantz, H., 1994, *A Robust Method to Estimate the Maximal Lyapunov Exponent of a Time Series*, Physical Letters A, vol. 185(1), s. 77–87.
- Kantz, H., Schreiber, T., 2004, *Nonlinear Time Series Analysis*, Cambridge University Press, Cambridge.
- Kennel, M.B., Brown, R., Abarbanel, H.D.I., 1992, *Detecting Embedding Dimension for Phase Space Reconstruction Using a Geometrical Construction*, Physical Review A, vol. 45.
- Largest Lyapunov Exponent with Rosenstein's Algorithm*, <http://www.mathworks.com/matlabcentral/fileexchange/38424-largest-lyapunov-exponent-with-rosenstein-s-algorithm>.
- Packard, N.H., Crutchfield, J.P., Farmer, J.D., Shaw R.S., 1980, *Geometry from a Time Series*, Physical Review Letters, vol. 45, no. 9, s. 712–716.
- Peitgen, H., Jürgens, H., Saupe, D., 1996, *Granice chaosu. Fraktale*, Wydawnictwo Naukowe PWN, Warszawa.
- Peters, E.E., 1997, *Teoria chaosu a rynki kapitałowe*, WiG_Press, Warszawa.
- Rosenstein, T., Collins, J.J., De Luca, C.J., 1993, *A Practical Method for Calculating Largest Lyapunov Exponents from Small Data Sets*, Physica D: Nonlinear Phenomena, vol. 65, no. 1, s. 117–134.
- Schuster, H.G., 1995, *Chaos deterministyczny*, Wydawnictwo Naukowe PWN, Warszawa.
- Shapour, M., 2017, *LYAPROSEN: MATLAB Function to Calculate Lyapunov Exponent*, <https://ideas.repec.org/c/boc/bocode/t741502.html>.
- Wolf, A., Swift, J., Swinney, H., Vastano, J., 1985, *Determining Lyapunov Exponents from a Time Series*, Physica D, vol. 16, s. 285–317.