Dowody twierdzeń wspierane obliczeniami przeprowadzonymi na komputerze

Daniel WILCZAK, Kraków*

Wprowadzenie

Jest to tekst związany z odczytem wygłoszonym na LIX Szkole Matematyki Poglądowej, *Matematyka i komputery*, Wola Ducka, luty 2018.

Redakcja

Programy komputerowe, pomimo wypracowywania coraz bardziej efektywnych form testowania, na ogół nie są wolne od defektów. Szczególnie niebezpiecznymi defektami są te, które nie są sygnalizowane jako błędy czy też ostrzeżenia zarówno w czasie kompilacji, jak i wykonywania programu – na przykład przepełnienie (wynik operacji przekracza maksymalną/minimalną wartość dopuszczaną przez stosowany typ liczbowy). Doprowadziły one, na przykład, do katastrofy rakiety nośnej Ariane 5 w 1996, co potwierdził raport [3] komisji powołanej do zbadania przyczyny tej eksplozji.

Rozważmy następujący przykład pochodzący od S.M. Rumpa [4] z 1998 roku.

Przykład 1. *Oblicz na komputerze wartość funkcji*

$$f(x,y) = 333.75y^6 + x^2(11x^2y^2 - y^6 - 121y^4 - 2) + 5.5y^8 + x/(2y) \tag{1}$$

dla $x = x_0 := 77617$ *i* $y = y_0 := 33096$.

Wyniki eksperymentu numerycznego są przedstawione w poniższej tabeli.

Użyte narzędzie	*Obliczona wartość*
Open Office 3.0 Calc	1.1726039
C++ float (32 bity)	$-6.338253001141147 \cdot 10^{29}$
C++ `double` (64 bity)	$-1.1805916207174113 \cdot 10^{21}$
C++ long `double` (80 bitów)	$5.7646 \cdot 10^{17}$
Mathematica	$-1.18059 \cdot 10^{21}$
GNU multiprecision library (140 bitów)	-0.82739605994682137
Mathematica – wynik dokładny	$-54767/66192 \approx -0.82739605994682137$

Przykład ten ilustruje efekt skrajnego kasowania bitów znaczących podczas dodawania liczb zmiennoprzecinkowych. Dwa składniki sumy

$$\begin{aligned} T_1 &= 5.5y_0^8 \\ &= +7917111340668961361101134701524942848, \\ T_2 &= 333.75y_0^6 + x_0^2(11x_0^2y_0^2 - y_0^6 - 121y_0^4 - 2) \\ &= -7917111340668961361101134701524942850 \end{aligned}$$

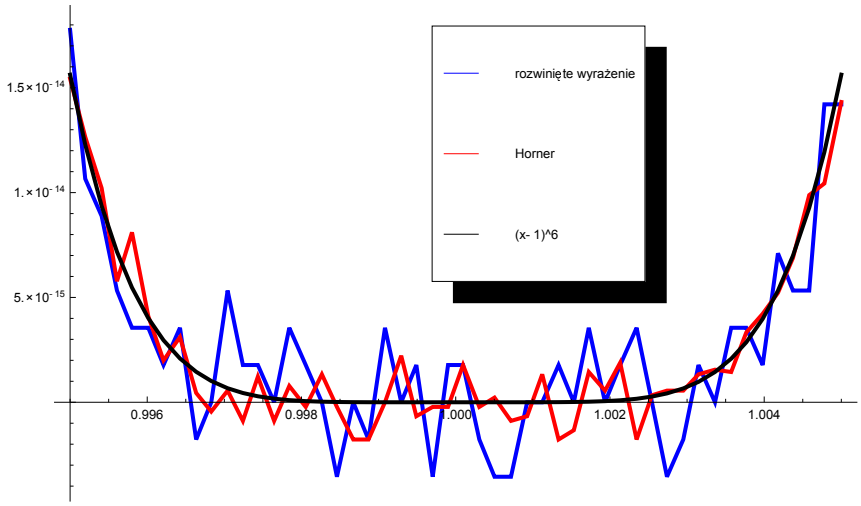
mają bardzo duże moduły, a ich suma wynosi zaledwie -2. Przy obliczeniach np. na liczbach typu `double` mamy do dyspozycji jedynie 52 bity mantysy, co przekłada się na około $52 \log_{10} 2 \approx 16$ dziesiętnych cyfr znaczących.

Przykład 2. *Narysuj wykres wielomianu* $f(x) = (x-1)^6$ *w otoczeniu* $x = 1$ *używając trzech reprezentacji*

$$\begin{aligned} f(x) &= (x-1)^6, \\ &= x^6 - 6x^5 + 15x^4 - 20x^3 + 15x^2 - 6x + 1, \quad \textit{(rozwinięcie)} \\ &= 1 + x(-6 + x(15 + x(-20 + x(15 + x(-6 + x))))). \quad \textit{(Horner)} \end{aligned}$$

Wynik eksperymentu przedstawiono na rysunku 1. W każdym z przypadków wykres sporządzono obliczając przybliżoną wartość wyrażenia w wybranych punktach przedziału, a następnie łącząc liniami (przybliżone) punkty na wykresie. Jak widać, obliczone wartości mogą być ujemne. Powodem różnic jest między innymi brak łączności dodawania i odejmowania w arytmetyce zmiennoprzecinkowej.

*Wydział Matematyki i Informatyki, Uniwerystet Jagielloński
wilczak@ii.uj.edu.pl



Rys. 1. Przybliżony wykres wielomianu $f(x) = (x-1)^6$ dla trzech różnych reprezentacji wyrażenia.

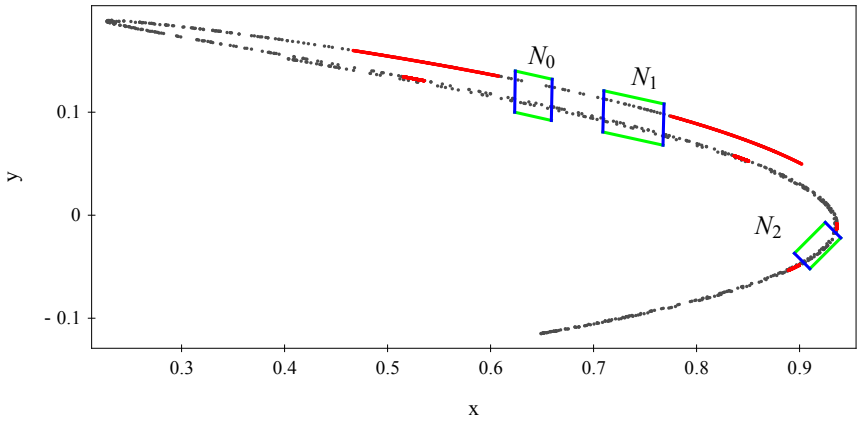
Powyższe przykłady powinny wzbudzić naszą czujność wobec określenia „komputerowo wspierany dowód twierdzenia". W dalszej części artykułu postaram się uzasadnić, że umiejętne kontrolowanie błędów wynikających z operacji wykonywanych na liczbach zmiennoprzecinkowych daje nam możliwość weryfikowania silnych nierówności. Łącząc to z dobrze dobranymi abstrakcyjnymi twierdzeniami dostajemy skuteczne narzędzia pozwalające na badanie konkretnych modeli matematycznych (skupiam się tutaj na układach dynamicznych z czasem dyskretnym lub ciągłym). Ogólne rozważania będą zilustrowane dwoma (elementarnymi z dzisiejszej perspektywy) przykładami – istnienie atraktora i dynamiki symbolicznej w odwzorowaniu Rösslera [9] oraz dla klasycznego układu równań różniczkowych Rösslera [8].

Chaos w odwzorowaniu Rösslera

Odwzorowanie Rösslera [9] to dyfeomorfizm płaszczyzny $\mathcal{R} = (\mathcal{R}_1, \mathcal{R}_2) : \mathbb{R}^2 \to \mathbb{R}^2$ określony formułą

$$(2) \qquad \mathcal{R}_1(x,y) = 3.8x(1-x) - 0.1y, \qquad \mathcal{R}_2(x,y) = 0.2(y-1.2)(1-1.9x).$$

W swojej pracy magisterskiej [11] pokazałem, że odwzorowanie to posiada zwarty i spójny atraktor o skomplikowanej strukturze – rysunek 2.



Rys. 2. Obserwowany atraktor dla odwzorowania (2) oraz zbiory N_i, $i = 0, 1, 2$, które posłużyły do konstrukcji dynamiki symbolicznej.

Istnienie atraktora jest konsekwencją następującego prostego lematu.

Lemat 1. *Zbiór* $W = [\,0.01\,,\,0.99\,] \times [-0.33\,,\,0.27\,]$ *jest dodatnio niezmienniczy względem* $\mathcal{R}$.

Dowód. Dla $(x, y) \in W$ mamy

$$\begin{aligned}
&\mathcal{R}_1(x,y) \leqslant 3.8 \cdot (0.5)^2 + 0.1 \cdot 0.33 = 0.983 < 0.99,\\
&\mathcal{R}_1(x,y) \geqslant 3.8 \cdot 0.99 \cdot 0.01 - 0.1 \cdot 0.27 = 0.01062 > 0.01,\\
&\mathcal{R}_2(x,y) \leqslant 0.2(-1.53)(1 - 1.9 \cdot 0.99) = 0.269586 < 0.27,\\
&\mathcal{R}_2(x,y) \geqslant 0.2(-0.33 - 1.2)(1 - 1.9 \cdot 0.01) = -0.300186 > -0.33.
\end{aligned}$$

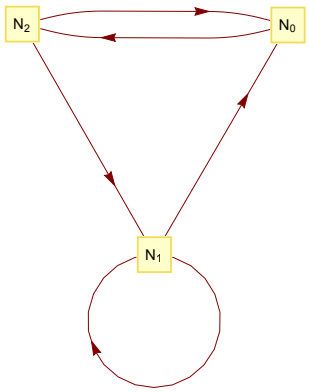
□

Głównym wynikiem pracy [11] jest następujące twierdzenie.

Twierdzenie 1. *Każda ścieżka* $(N_{i_j})_{j\in\mathbb{Z}} \in \{N_0, N_1, N_2\}^{\mathbb{Z}}$ *w grafie przedstawionym na rysunku 3 jest realizowana przez pewną trajektorię* $(u_j)_{j\in\mathbb{Z}}$ *dla* $\mathcal{R}^3$. *Dokładniej, dla* $j \in \mathbb{Z}$ *mamy*

$$u_{j+1} = \mathcal{R}^3(u_j), \quad u_j \in N_{i_j}.$$

Ponadto, jeśli ścieżka $(N_{i_j})_{j\in\mathbb{Z}}$ *jest cyklem, to odpowiadająca jej trajektoria* $(u_j)_{j\in\mathbb{Z}}$ *może być wybrana jako okresowa dla* $\mathcal{R}^3$ *o takim samym okresie podstawowym, jak długość cyklu w grafie.*



Rys. 3. Graf dynamiki symbolicznej dla odwzorowania Rösslera (2).

Jak widać istnieją trajektorie dla $\mathcal{R}^3$, które odwiedzają zbiory N_0, N_1, N_2 w sposób określony przez (dowolną) ścieżkę w grafie. O takich układach mówi się, że są semisprzężone z **dynamiką symboliczną** na k symbolach, gdzie k jest liczbą rozłącznych zbiorów, pomiędzy którymi trajektorie mogą wędrować. W Twierdzeniu 1 wykazano semisprzężenie z dynamiką symboliczną na trzech symbolach.

Przy ustalonej liczbie symboli k pewną miarą skomplikowania dynamiki jest liczba różnych możliwych ścieżek w grafie (skierowanym). Jest ona oczywiście największa, gdy graf jest pełny (ma wszystkie możliwe krawędzie oraz pętelki w wierzchołkach). Mówimy wtedy o semisprzężeniu z **pełną dynamiką symboliczną na** k **symbolach**. Liczba różnych ścieżek w grafie może (ale nie musi) rosnąć wraz ze zwiększaniem liczby symboli k.

Semisprzężenie odwzorowania f z dynamiką symboliczną oznacza w szczególności, że dynamika f jest co najmniej tak skomplikowana, jak wędrowanie po grafie dynamiki symbolicznej. W szczególności tzw. entropia topologiczna f, która jest pewną miarą skomplikowania dynamiki, jest co najmniej taka, jak dla odwzorowania „wędrowania po grafie". O entropii topologicznej możemy też (nieformalnie) myśleć jako o współczynniku wykładniczego wzrostu liczby różnych orbit okresowych wraz z okresem podstawowym.

W przypadku odwzorowania $\mathcal{R}^3$ liczba różnych cykli w grafie o długości m (a co za tym idzie liczba różnych trajektorii okresowych dla $\mathcal{R}^3$ o okresie m) rośnie

wykładniczo szybko wraz z m. Okazuje się, że w ogólnym przypadku logarytm największej na moduł wartości własnej macierzy sąsiedztwa grafu określa współczynnik wykładniczego wzrostu liczby różnych cykli o ustalonej długości. W przypadku odwzorowania $\mathcal{R}^3$ macierz sąsiedztwa grafu to

$$A = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix},$$

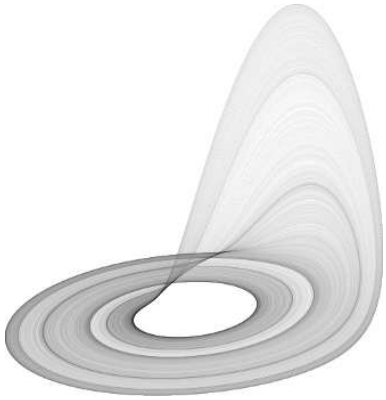
a wartość własna A o największym module to $\lambda := \frac{1}{2}(\sqrt{5}+1)$. Zatem entropia topologiczna $h_{top}(\mathcal{R}^3) \geqslant \ln \lambda \approx 0.481212$.

Chaos w układzie Rösslera

Otto Rössler [8] zaproponował układ równań różniczkowych zwyczajnych

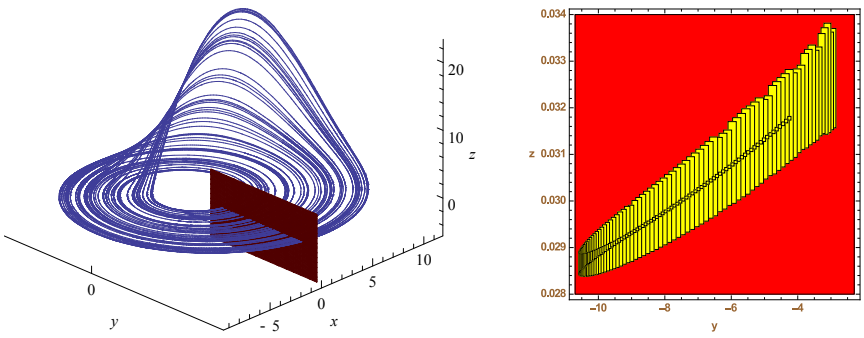
$$\begin{cases} x' = -y - z \\ y' = x + by \\ z' = b + z(x - a) \end{cases}, \tag{3}$$

w którym dla pewnych „klasycznych" wartości parametrów $b = 0.2$, $a = 5.7$ obserwowane jest istnienie chaotycznego atraktora – zobacz rysunek 4.



Rys. 4. Obserwowany atraktor w układzie (3) dla parametrów $b = 0.2$, $a = 5.7$.

Określamy sekcję Poincarégo $\Pi = \{(0, y, z) : y, z \in \mathbb{R}, x' > 0\}$. Zbiór Π jest podzbiorem płaszczyzny $\{x = 0\}$ składającym się z punktów spełniających warunek $x' = -y - z > 0$. Oznacza to, że składowa $x(t)$ trajektorii układu (3) przy przecięciu ze zbiorem Π zmienia znak z minus na plus. Fragment sekcji Π oraz przykładowa trajektoria układu są przedstawione na rysunku 5.



Rys. 5. Z lewej – przykładowa trajektoria układu (3) dla parametrów $b = 0.2$, $a = 5.7$ oraz fragment sekcji Poincarégo Π. Z prawej – obszar pułapka B (na czerwono) i oszacowanie na jego obraz $\mathcal{P}(B)$ (na żółto).

Możemy teraz określić odwzorowanie Poincarégo $\mathcal{P} : \Pi \to \Pi$. Dla $x \in \Pi$ definiujemy $\mathcal{P}(x)$ jako punkt pierwszego powrotu trajektorii punktu x do zbioru Π, o ile taki punkt istnieje. W przeciwnym przypadku x nie jest w dziedzinie $\mathcal{P}$. Z założenia transwersalności przecięcia $x' > 0$ można wywnioskować, że odwzorowanie $\mathcal{P}$ jest różniczkowalne na swojej dziedzinie.

Komputerowo wspierany dowód istnienia dynamiki symbolicznej oraz nieskończenie wielu orbit okresowych o dowolnie wysokich okresach podstawowych w układzie (3) podał Zgliczyński [12]. W pracy [10] wynik ten został rozszerzony do następującego twierdzenia.

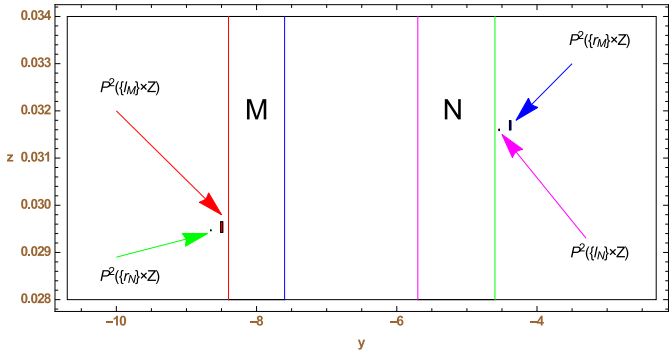
Twierdzenie 2. *Określamy podzbiory* Π *(pomijamy współrzędną* $x = 0$*)*

$$B = [l_B, r_B] \times Z, \qquad M = [l_M, r_M] \times Z, \qquad N = [l_N, r_N] \times Z,$$

gdzie $l_B = -10.7$, $r_B = -2.3$, $l_M = 8.4$, $r_M = -7.6$, $l_N = -5.7$, $r_N = -4.6$, $Z = [0.028, 0.034]$. *Wtedy*

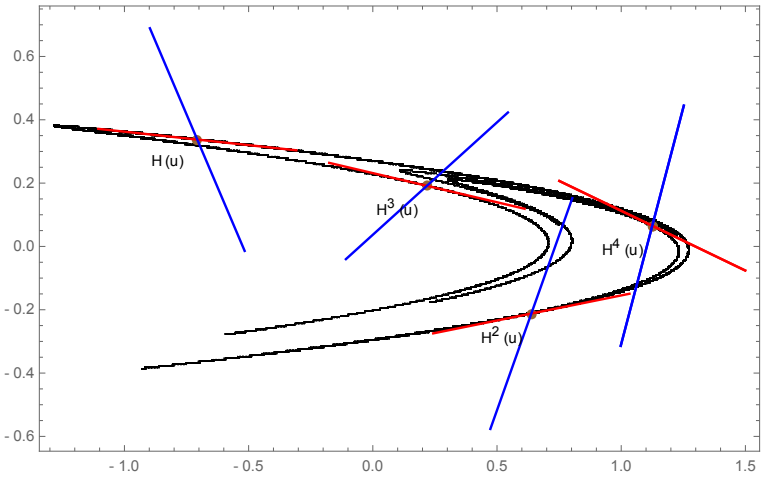
- *zbiór* B *jest dodatnio niezmienniczy dla* $\mathcal{P}$ *(czyli* $\mathcal{P}(B) \subset B$*), co implikuje istnienie zwartego, spójnego atraktora* $\mathcal{A} = \bigcap_{i>0} \mathcal{P}^i(B)$ *w układzie* (3) *jako przecięcia zstępującego ciągu zbiorów zwartych i spójnych;*
- *maksymalny podzbiór niezmienniczy* $\mathcal{H} = \mathrm{Inv}(P^2, N \cup M)$ *jest jednostajnie hiperboliczny dla* $\mathcal{P}^2$*;*
- *dynamika* $\mathcal{P}^2$ *w zawężeniu do* $\mathcal{H}$ *jest sprzężona z pełną dynamiką symboliczną na dwóch symbolach. W szczególności zbiór* $\mathcal{H}$ *zawiera przeliczalną liczbę orbit okresowych o dowolnie dużych okresach podstawowych.*

Zbiór dodatnio niezmienniczy B oraz *oszacowanie* na jego obraz $\mathcal{P}(B)$ przedstawiono na rysunku 5. Zbiory N, M zostały pokazane na rysunku 6.



Rys. 6. Zbiory N i M użyte do konstrukcji dynamiki symbolicznej oraz oszacowanie na obraz ich wybranych krawędzi poprzez $\mathcal{P}^2$.

Pojęcie hiperboliczności zbioru niezmienniczego jest uogólnieniem znanego pojęcia hiperboliczności punktu stałego, czy też hiperboliczności odwzorowania liniowego – wartości własne różniczki odwzorowania w punkcie stałym nie leżą na okręgu jednostkowym. W takim przypadku podprzestrzeń styczna w punkcie stałym rozpada się na dwie podprzestrzenie niezmiennicze. Jedna odpowiada wszystkim wartościom własnym o module mniejszym od jeden, w której wektory są wykładniczo szybko skracane przez różniczkę odwzorowania. W tej drugiej, odpowiadającej wartościom własnym spoza koła jednostkowego, wektory są wykładniczo szybko wydłużane.



Rys. 7. Hiperboliczna orbita o okresie 4 dla odwzorowania Hénona wraz z rozbiciem hiperbolicznym. Kierunki niebieskie są cyklicznie przekształcane na siebie przez różniczkę odwzorowania, a wektory w nich są wykładniczo szybko skracane. Analogicznie w podprzestrzeniach czerwonych wektory są wykładniczo szybko wydłużane.

Podobnie wygląda to dla hiperbolicznych orbit okresowych. W każdym punkcie takiej orbity mamy wyróżnione kierunki wykładniczego skracania i wydłużania wektorów z przestrzeni stycznej. Ponadto te kierunki są cyklicznie przekształcane na siebie przez różniczkę odwzorowania w kolejnych punktach trajektorii – zobacz rysunek 7.

Idąc dalej, możemy uogólnić pojęcie hiperboliczności na dowolne zbiory niezmiennicze – rozbicie na kierunki wykładniczego rozciągania i ściągania odbywa się wzdłuż każdej trajektorii tego zbioru. Ponadto prędkości wykładniczego ściągania i rozciągania są wspólne i jednostajnie odseparowane od 1 – niezależnie od wyboru trajektorii.

Zgliczyński [12] podał geometryczne warunki (nazywane dzisiaj relacjami nakrywającymi), które implikują istnienie dynamiki symbolicznej dla pewnego ciągłego przekształcenia płaszczyzny. W przypadku odwzorowania Poincarégo $\mathcal{P}$ z Twierdzenia 2 sprowadzają się one do pięciu nierówności:

$$
\begin{array}{ll}
\pi_y P^2(y,z) < l_M & \text{dla } (y,z) \in \{l_M\} \times Z, \\
\pi_y P^2(y,z) > r_N & \text{dla } (y,z) \in \{r_M\} \times Z, \\
\pi_y P^2(y,z) < l_M & \text{dla } (y,z) \in \{r_N\} \times Z, \\
\pi_y P^2(y,z) > r_N & \text{dla } (y,z) \in \{l_N\} \times Z, \\
\pi_z P^2(y,z) \in \operatorname{int} Z & \text{dla } (y,z) \in N \cup M.
\end{array} \tag{4}
$$

Symulacja numeryczna (rysunek 6) daje silną przesłankę, że te nierówności są spełnione. Dowód hiperboliczności zbioru niezmienniczego sprowadza się do sprawdzenia dodatniej określoności macierzy $D\mathcal{P}^2(x)^T Q D\mathcal{P}^2(x) - Q$, dla wszystkich $x \in N \cup M$, gdzie Q jest dowolną, ustaloną nieosobliwą macierzą diagonalną [10].

Analogiczne warunki można sformułować dla odwzorowania (2) i dotyczą one obrazów krawędzi równoległoboków N_0, N_1, N_2, jak pokazano na rysunku 2. Odwzorowanie Rösslera (2) zadane jest jawnym wzorem. Dlatego sprawdzanie afinicznej nierówności $Ax + By + C > 0$ na krawędzi $[p,q]$ jednego ze zbiorów N_i sprowadza się do badania znaku wielomianu jednej zmiennej stopnia co najwyżej osiem

$$[0,1] \ni t \to \left\langle [A,B]; \mathcal{R}^3(pt + (1-t)q) \right\rangle + C \in \mathbb{R}.$$

Jest to zadanie teoretycznie możliwe do wykonania na papierze, niemniej bardzo uciążliwe.

Analiza staje się bardzo trudna w przypadku odwzorowania Poincarégo $\mathcal{P}$ i warunków (4). Tutaj pojawia się miejsce, gdzie można wykorzystać moc obliczeniową komputerów. Potrzebne są algorytmy, które będą obliczały oszacowania obrazów zbiorów przez funkcje i ich pochodne. Podstawowymi narzędziami ścisłej analizy numerycznej są *arytmetyka przedziałowa* oraz tzw. *algorytmiczne różniczkowanie*, które omówię w kolejnych rozdziałach.

Arytmetyka przedziałowa

Komputery w swojej naturze są skończone – potrafią zapamiętać skończenie wiele danych i wykonać skończoną liczbę operacji na nich. Nie da się zakodować w pamięci komputera liczb rzeczywistych, a tym bardziej $2^{\mathbb{R}}$. Dlatego potrzebny jest wybór rozsądnej, dostatecznie bogatej klasy podzbiorów $\mathbb{R}^n$, której elementy z jednej strony można łatwo kodować w pamięci komputera, a z drugiej można efektywnie wykonywać na nich operacje mnogościowe, czy też oszacowywać ich obrazy przez funkcje elementarne. Jednym z możliwych wyborów jest klasa przedziałów domkniętych [7] i ogólnie kostek (iloczynów kartezjańskich przedziałów) w $\mathbb{R}^n$. Będziemy rozważać tylko przedziały zwarte i oznaczać je przez

$$[a] = [\underline{a}, \overline{a}] = \{x \in \mathbb{R} : \underline{a} \leqslant x \leqslant \bar{a}\}.$$

Zbiór wszystkich zwartych przedziałów będziemy oznaczać przez

$$\mathbb{I} = \{[\underline{a}, \overline{a}] : \underline{a}, \overline{a} \in \mathbb{R},\ \underline{a} \leqslant \overline{a}\}.$$

Dla ograniczonego i niepustego podzbioru $S \subset \mathbb{R}^n$ oznaczamy otoczkę przedziałową zbioru S przez

$$[S]_I = \bigcap \{[u] : [u] \in \mathbb{I}^n, S \subset [u]\} \in \mathbb{I}^n,$$

przy czym w powyższym utożsamiamy elementy

$$\mathbb{I}^n \ni ([x_1], \ldots, [x_n]) \to [x_1] \times \cdots \times [x_n] \in 2^{\mathbb{R}^n}.$$

Elementy $\mathbb{I}^n$ będziemy nazywać **wektorami przedziałowymi**. Elementy $\mathbb{I}^n$ jako zbiory z dokładnością do powyższego utożsamienia, w naturalny sposób mają zdefiniowane relacje

$$=, \qquad \subseteq, \qquad \subset, \qquad \neq, \qquad \not\subset .$$

Ponadto możemy zdefiniować działania

$$\sqcap : \mathbb{I}^n \times \mathbb{I}^n \ni ([a], [b]) \to [a] \cap [b] \in \mathbb{I}^n \cup \{\emptyset\},$$
$$\sqcup : \mathbb{I}^n \times \mathbb{I}^n \ni ([a], [b]) \to [\,[a] \cup [b]\,]_I \in \mathbb{I}^n.$$

Działania w arytmetyce przedziałowej.

Niech $[a] = [\,\underline{a}, \overline{a}\,]$ oraz $[b] = \left[\,\underline{b}, \overline{b}\,\right]$ będą dwoma niepustymi przedziałami.

Definicja 1. *Określamy rozszerzenie standardowych działań arytmetycznych dla liczb rzeczywistych $\star \in \{+, -, *, \div\}$ na przedziały $[a], [b] \in \mathbb{I}$ za pomocą*

$$[a] \star [b] = \{x \star y : x \in [a],\ y \in [b]\}\,,$$

przy czym dla dzielenia zakładamy dodatkowo, że $0 \notin [b]$.

Zauważmy, że:

- każde działanie elementarne na przedziałach można zrealizować za pomocą skończonej liczby porównań oraz działań na ich końcach:

$$\begin{aligned}
[\,\underline{a}, \overline{a}\,] + \left[\,\underline{b}, \overline{b}\,\right] &= [\underline{a} + \underline{b}, \overline{a} + \overline{b}],\\
[\,\underline{a}, \overline{a}\,] - \left[\,\underline{b}, \overline{b}\,\right] &= [\underline{a} - \overline{b}, \overline{a} - \underline{b}],\\
[\,\underline{a}, \overline{a}\,] * \left[\,\underline{b}, \overline{b}\,\right] &= [\min\{\underline{a} * \underline{b}, \overline{a} * \underline{b}, \underline{a} * \overline{b}, \overline{a} * \overline{b}\}, \max\{\underline{a} * \underline{b}, \overline{a} * \underline{b}, \underline{a} * \overline{b}, \overline{a} * \overline{b}\}],\\
[a] \div \left[\,\underline{b}, \overline{b}\,\right] &= [a] * [1/\overline{b}, 1/\underline{b}], \quad \text{o ile } 0 \notin \left[\,\underline{b}, \overline{b}\,\right];
\end{aligned}$$

- nie ma rozdzielności mnożenia względem dodawania, np. dla $[a] = [-1, 1]$, $b = [2, 2]$ mamy

$$\begin{aligned}
[a] * [b] - [a] * [b] &= [-1, 1] * [2, 2] - [-1, 1] * [2, 2]\\
&= [-2, 2] - [-2, 2] = [-4, 4]\\
[a] * ([b] - [b]) &= [-1, 1] * [0, 0] = [0, 0].
\end{aligned}$$

- dla dowolnych przedziałów $[a], [b], [c]$ prawdziwa jest inkluzja $[a] * ([b] + [c]) \subset [a] * [b] + [a] * [c]$.

W podobny sposób można zdefiniować rozszerzenie funkcji elementarnych

$$\sin, \cos, \sqrt{\ }, \exp, \log, \ldots$$

na argumenty przedziałowe; na przykład

$$\sin([x]) = [\inf\{\sin(a) : a \in [x]\}, \sup\{\sin(a) : a \in [x]\}].$$

Definicja 2. *Funkcję $f : \mathbb{R}^n \to \mathbb{R}$, którą można wyrazić za pomocą skończonej liczby złożeń funkcji i działań elementarnych, nazywamy* funkcją prostą. *Dla funkcji prostej $f : \mathbb{R}^n \to \mathbb{R}$ z ustaloną reprezentacją (skończona liczba złożeń funkcji i operacji elementarnych) przez $[f]([x_1], [x_2], \ldots, [x_n])$ będziemy oznaczać ewaluację funkcji f w arytmetyce przedziałowej.*

Zwykle reprezentacja funkcji f jest ustalona i nie prowadzi to do nieporozumień.

Przykład 3. *Oszacowanie $[f]([x])$ może zależeć od reprezentacji f. Weźmy na przykład funkcję $f(x) = (x-1)^2 - 1 = x(x-2) = x^2 - 2x$ oraz $[x] = [0, 1]$. Wtedy*

$$\begin{aligned}
([x] - 1)^2 - 1 &= [-1, 0]^2 - 1 = [-1, 0],\\
[x] * ([x] - 2) &= [0, 1] * ([0, 1] - 2) = [0, 1] * [-2, -1] = [-2, 0],\\
[x]^2 - 2[x] &= [0, 1]^2 - 2[0, 1] = [0, 1] - [0, 2] = [-2, 1].
\end{aligned}$$

Niezależnie od reprezentacji wyrażenia otrzymany przedział zawiera obraz odcinka $[0, 1]$ przez funkcję f. Przykład ten sugeruje, że należy minimalizować liczbę wystąpień zmiennych w wyrażeniach (tzw. *dependency problem*).

Twierdzenie 3. ([7]) *Jeśli* $f : \mathbb{R}^n \to \mathbb{R}$ *jest funkcją prostą,* $[u] \in \mathbb{I}^n$ *oraz* $[f]([u])$ *istnieje, to*

$$f([u]) \subset [f]([u]).$$

W Definicji 1 przyjęliśmy założenie, że potrafimy dokładnie obliczać sumę, różnicę, iloczyn i iloraz dwóch przedziałów za pomocą operacji na ich końcach. Można ograniczyć się do przedziałów o końcach wymiernych. Klasa takich przedziałów jest domknięta ze względu na operacje sumy, różnicy, iloczynu i ilorazu. Niestety, koszt obliczeń na liczbach wymiernych jest na ogół nieakceptowalny — w praktycznych zastosowaniach pojawiają nieskracalne ilorazy bardzo dużych liczb całkowitych. Kompromis między szybkością obliczeń i dokładnością wyników realizują liczby zmiennoprzecinkowe. Klasa ta wprawdzie nie jest domknięta ze względu na operacje arytmetyczne, ale w standardzie IEEE 754 [6] bardzo precyzyjnie określono sposób rzutowania (zaokrąglania) wyników do liczb zmiennoprzecinkowych.

Standard IEEE 754

Standard IEEE 754 [6] określa między innymi

- sposób reprezentacji liczb zmiennoprzecinkowych oraz
- zasady wykonywania obliczeń na liczbach zmiennoprzecinkowych.

W obliczeniach przeprowadzanych na komputerze najczęściej korzysta się z liczb typu `double`. Każda liczba typu `double` jest zakodowana w **64-bitowym** ciągu. Prostą konsekwencją jest fakt, że w typie `double` możemy zakodować co najwyżej 2^{64} liczb rzeczywistych (w istocie znacznie mniej). Oznaczmy zbiór wszystkich liczb reprezentowalnych w formacie `double` przez $\widehat{\mathbb{R}}$.

Rounding

Standard IEEE 754 określa również kilka sposobów zaokrąglania liczb rzeczywistych do liczb reprezentowalnych – podamy tylko dwa najważniejsze z punktu widzenia dalszej części artykułu. Są to

- **roundUp**, czyli zaokrąglanie w górę:

$$\uparrow : \mathbb{R} \ni x \to \min\{u \in \overline{\mathbb{R}} : x \leqslant u\} \in \widehat{\mathbb{R}},$$

- **roundDown**, czyli zaokrąglanie w dół:

$$\downarrow : \mathbb{R} \ni x \to \max\{u \in \overline{\mathbb{R}} : u \leqslant x\} \in \widehat{\mathbb{R}}.$$

Działania elementarne

Standard IEEE 754 określa również dokładność z jaką procesor musi wykonać operacje arytmetyczne dla: **dodawania, odejmowania, mnożenia, dzielenia i pierwiastkowania**. W szczególności, jeśli $a, b \in \widehat{\mathbb{R}}$, $\star \in \{+, -, *, \div\}$ oraz procesor jest w trybie zaokrąglania $\circlearrowleft \in \{\uparrow, \downarrow\}$, to

$$\begin{aligned} |a \circlearrowleft (\star) b - a \star b| &\leqslant |a \star b| \varepsilon_M \\ a \downarrow (\star) b \leqslant a \star b &\leqslant a \uparrow (\star) b \end{aligned}$$

gdzie ε_M jest precyzją w danej reprezentacji i np. dla typu `double` wynosi $\varepsilon_M = 2^{-53}$. W powyższym, przez $a \circlearrowleft (\star) b$ rozumiemy wynik działania arytmetycznego $a \star b$ wykonanego przez procesor w trybie zaokrąglania $\circlearrowleft$.

Przedziały reprezentowalne

Oznaczenie: Zbiór przedziałów, których końce są liczbami reprezentowalnymi (czyli ze zbioru $\widehat{\mathbb{R}}$) nazywamy przedziałami reprezentowalnymi i oznaczamy $\widehat{\mathbb{I}}$.

Arytmetykę przedziałów rzeczywistych możemy zawęzić do przedziałów reprezentowalnych z uwzględnieniem funkcji zaokrąglających. Dla operacji elementarnych określonych w standardzie IEEE 754, czyli $\{+, -, *, \div, \sqrt{}\}$ oraz $[a], [b] \in \widehat{\mathbb{I}}$ określamy

$$[\underline{a}, \overline{a}] \hat{+} [\underline{b}, \overline{b}] = [\underline{a} \downarrow (+)\underline{b}, \overline{a} \uparrow (+)\overline{b}] \in \widehat{\mathbb{I}},$$
$$[\underline{a}, \overline{a}] \hat{-} [\underline{b}, \overline{b}] = [\underline{a} \downarrow (-)\overline{b}, \overline{a} \uparrow (-)\underline{b}] \in \widehat{\mathbb{I}}.$$

Podobnie dla mnożenia, dzielenia i pierwiastkowania. Pozostałe funkcje elementarne, które nie są wspierane standardem, mogą być zrealizowane za pomocą istniejących operacji, chociaż w niejednoznaczny i zależny od implementacji sposób. W dalszej części artykułu będę używał symboli działań arytmetycznych dla przedziałów reprezentowalnych z pominięciem znaku$\hat{}$ (z kontekstu będzie wiadomo, czy należy zastosować zaokrąglanie).

Przykład 4. *Funkcję wykładniczą można zrealizować w arytmetyce przedziałów reprezentowalnych w nastepujący sposób.*
Niech $[e]$ *będzie najmniejszym w sensie inkluzji przedziałem reprezentowalnym zawierającym stałą Eulera. Jeśli* $z \in [0, 1) \cap \widehat{\mathbb{R}}$*, to*

$$\exp(z) = \sum_{i=0}^{20} \frac{z^i}{i!} + \frac{z_*^{21}}{(21)!}$$

dla pewnego $z_* \in [0, z]$*. Wykonując obliczenia na komputerze powyższe wyrażenie ewaluujemy na przykład tak:*

$$\exp([z, z]) \in \sum_{i=0}^{20} \frac{[z, z]^i}{i!} + \frac{[0, z]^{21}}{(21)!}$$

przy czym wielomian może być ewaluowany za pomocą schematu Hornera.

Jeżeli $z \notin [0, 1)$ *to* $z = p + y$*, gdzie* p *jest liczbą całkowitą, a* $y \in [0, 1)$*. Wtedy* $\exp(z) \in [e]^p * \exp([y])$*. Wyrażenie* $[e]^p$ *jest zwykłym iloczynem przedziałów.*

Dla dowolnego przedziału reprezentowalnego $[x] = [\underline{x}, \overline{x}]$*, korzystając z monotoniczności funkcji wykładniczej można określić*

$$\exp([x]) := \exp([\underline{x}, \underline{x}]) \sqcup \exp([\overline{x}, \overline{x}]).$$

Przykład 5. *Wartość wyrażenia* (1) *oszacowana w arytmetyce przedziałów reprezentowalnych to* $[-5.9029581035870565, 4.7223664828696463] \cdot 10^{21}$.

Otrzymany przedział oczywiście zawiera dokładny wynik, a jego (nie)dokładność sugeruje, że pojawiły się problemy numeryczne przy ewaluacji.

Algorytmiczne różniczkowanie

Poniżej przedstawię kilka podstawowych technik algorytmicznego różniczkowania, czyli sposobów numerycznego obliczania pochodnych funkcji. Najważniejsze cechy tej grupy algorytmów to

- obliczenie przybliżonej wartości pochodnej nie wymaga wyznaczania wzorów na pochodną (sic!),
- algorytmy można stosować do funkcji wielu zmiennych,
- funkcje nie muszą być dane jawnym wzorem – stosuje się je do funkcji uwikłanych czy rozwiązań równań różniczkowych,
- różniczkować możemy algorytmy (stąd nazwa metody).

Forward Differentiation

Zaczniemy od najprostszego przykładu pochodnej pierwszego rzędu dla funkcji jednej zmiennej.

Operacje i funkcje elementarne dla liczb (rzeczywistych lub zespolonych) rozszerzymy na odpowiadające im operacje na parach liczb zgodnie z dobrze

znanymi formułami rachunku różniczkowego. Pierwszy element pary (u, u') będzie odpowiadał za wartość pewnej funkcji w punkcie, natomiast drugi element tej pary – za pochodną tej funkcji w tym samym punkcie. Podamy dla przykładu kilka podstawowych operacji na parach:

- $(u, u') \pm (v, v') = (u \pm v, u' \pm v')$,
- $(u, u') * (v, v') = (uv, uv' + u'v)$,
- $(u, u')/(v, v') = (u/v, (u' - (u/v)v')/v)$,
- $\sin(u, u') = (\sin(u), \cos(u)u')$.

Jak to działa w praktyce? Najprościej zilustrować to na przykładzie.

Przykład 6. *Niech* $f(x) = \dfrac{(x+1)(x-2)}{x+3}$. *Obliczymy* $f(3)$ *oraz* $f'(3)$. *W tym celu*

- *każdą stałą* c, *która występuje we wzorze na* f *zastępujemy przez parę* $(c, 0)$ *(pochodna stałej po zmiennej niezależnej jest równa zero),*
- *każde wystąpienie zmiennej niezależnej zamieniamy na parę* $(x, 1)$.

Zastosowanie arytmetyki dla par daje:

$$\frac{((3,1)+(1,0)) * ((3,1)-(2,0))}{(3,1)+(3,0)} = \frac{(4,1)*(1,1)}{(6,1)} = \frac{(4,5)}{(6,1)} = \left(\frac{2}{3}, \frac{13}{18}\right).$$

Otrzymana na końcu para to $f(3)$ *oraz* $f'(3)$.

Prawda, że szybko? I bez wyliczania wzoru na f'.

Różniczkowanie funkcji wielu zmiennych

Wprowadzona wcześniej arytmetyka par może zostać rozszerzona na arytmetykę wektorów. Pierwszy współczynnik wektora reprezentuje wartość pewnego wyrażenia w punkcie, a kolejne współczynniki odpowiadają kolejnym pochodnym cząstkowym. Takie podejście pozwala nam na liczenie **jednocześnie** wszystkich (lub wybranych) pochodnych cząstkowych funkcji prostej.

Przykład 7. *Obliczymy wartość i pochodne cząstkowe* $f(x, y) = \frac{x+y+1}{xy-1}$ *w punkcie* $(2, -2)$. *Każde wystąpienie zmiennej* x *zastępujemy przez trójkę* $(x, 1, 0)$, y *przez* $(y, 0, 1)$ *i stałe przez* $(c, 0, 0)$. *Mamy*

$$\frac{(2,1,0)+(-2,0,1)+(1,0,0)}{(2,1,0)*(-2,0,1)-(1,0,0)} = \frac{(1,1,1)}{(-5,-2,2)} = \left(\frac{-1}{5}, \frac{-3}{25}, \frac{-7}{25}\right),$$

czyli $f(2,-2) = -\frac{1}{5}$, $\frac{\partial f}{\partial x}(2,-2) = -\frac{3}{25}$ *oraz* $\frac{\partial f}{\partial y}(2,-2) = -\frac{7}{25}$.

Metoda różniczkowania w przód jest zalecana do obliczania pochodnych funkcji $f\colon \mathbb{R}^n \to \mathbb{R}^m$, gdy $n \ll m$. Gdy n jest duże, pojawia się spory narzut obliczeniowy polegający na nadmiarowym propagowaniu zer w wektorach $(u_0, \ldots, u_n)$, np. dla funkcji 8 zmiennych $(x_1, \ldots, x_8)$, w której pierwszym działaniem jest dodawanie $x_1 + x_2$ musimy obliczyć

$$(x_1, 1, 0, 0, 0, 0, 0, 0) + (x_2, 0, 1, 0, 0, 0, 0, 0) = (x_1 + x_2, 1, 1, 0, 0, 0, 0, 0).$$

Istnieje grupa algorytmów (Backward Differentiation), w której ten narzut jest całkowicie wyeliminowany (pojawia się za to narzut pamięciowy), jednak ze względu na ograniczoną objętość tego artykułu pominiemy jej opis.

Szeregi Taylora funkcji jednej zmiennej

Wprowadzimy następujące oznaczenie:

$$f^{[k]}(x) = \frac{f^{(k)}(x)}{k!}$$

na k-ty współczynnik Taylora funkcji f w punkcie x. Okazuje się [5], że można łatwo obliczać współczynniki Taylora funkcji prostych rozszerzając wprowadzoną wcześniej arytmetykę dla par na arytmetykę wektorów $(u_0, u_1, \ldots, u_n)$, gdzie u_k jest wartością k-tego współczynnika Taylora pewnej funkcji w punkcie. Podamy

tutaj tylko kilka przykładowych formuł:

$$(f \pm g)^{[k]} = f^{[k]} \pm g^{[k]}$$

$$(f \cdot g)^{[k]} = \sum_{i=0}^{k} f^{[i]} \cdot g^{[k-i]} \text{ (wzór Leibniza)}$$

$$(f/g)^{[k]} = \frac{1}{g^{[0]}} \left(f^{[k]} - \sum_{i=0}^{k-1} (f/g)^{[i]} \cdot g^{[k-i]} \right) \text{ (przekształcony wzór Leibniza)}$$

$$(\exp(f))^{[k]} = \begin{cases} \exp(f^{[0]}) & \text{dla } k = 0 \\ \frac{1}{k} \sum_{i=1}^{k} i f^{[i]} \cdot (\exp(f))^{[k-i]} & \text{dla } k > 0 \end{cases}$$

Ostatni wzór można łatwo wyprowadzić wykorzystując tożsamość $g' = g \cdot f'$, gdzie $g = \exp(f)$, oraz wzór Leibniza. Dla $k > 0$ mamy

$$kg^{[k]} = \sum_{i=0}^{k-1} (f')^{[i]} g^{[k-1-i]} = \sum_{i=0}^{k-1} (i+1) f^{[i+1]} g^{[k-1-i]} = \sum_{i=1}^{k} i f^{[i]} g^{[k-i]}.$$

Przykład użycia tej arytmetyki zostanie podany w dalszej części artykułu, przy okazji omawiania zagadnień początkowych dla równań różniczkowych (Przykład 8).

Metoda Taylora dla równań różniczkowych

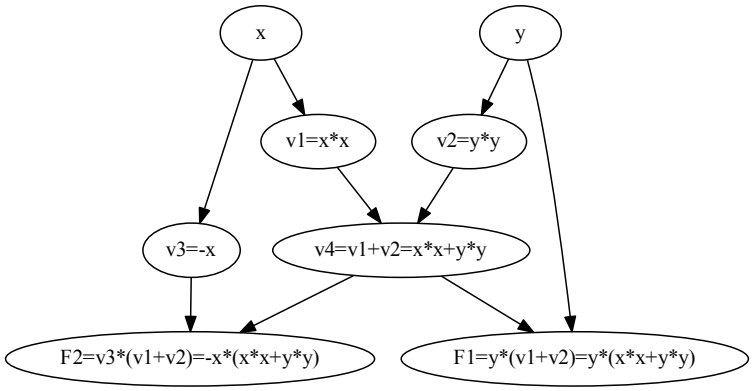
Rozważmy zagadnienie początkowe

$$x' = f(x), \qquad x(0) = x_0. \tag{5}$$

Oznaczmy $F = f \circ x : I \subset \mathbb{R} \to \mathbb{R}^n$, gdzie I jest pewnym przedziałem otwartym zawierającym zero. Zauważmy, że

$$x^{[n+1]}(0) = \frac{1}{n+1} F^{[n]}(0). \tag{6}$$

Powyższa tożsamość pozwala na iteracyjne wyznaczanie współczynników Taylora $x^{[n]}(0)$ **bez wyliczania wzorów na pochodne pola wektorowego**. Liczymy jedynie współczynniki Taylora dla F, która jest funkcją jednej zmiennej.



Rys. 8. Graf pola wektorowego z Przykładu 8.

Przykład 8. *Zilustrujemy metodę na przykładzie równania, które z jednej strony jest zadane nietrywialną formułą, a z drugiej da się rozwiązać analitycznie. Rozważmy równanie*

$$\begin{cases} \dot{x} = y(x^2 + y^2) \\ \dot{y} = -x(x^2 + y^2) \end{cases}$$

z warunkiem początkowym $(x_0, y_0) = (1, -1)$. Ustalamy reprezentację pola wektorowego f tak, jak przedstawiono na rysunku 8. Warunek początkowy ustala wartości $x^{[0]} = 1$ oraz $y^{[0]} = -1$ (tutaj i w dalszej części przykładu pomijamy

argument, który zawsze jest równy zero). Następnie ewaluujemy wyrażenie

$$\begin{aligned} v_1^{[0]} &= x^{[0]} * x^{[0]} = 1, \\ v_2^{[0]} &= y^{[0]} * y^{[0]} = 1, \\ v_3^{[0]} &= -x^{[0]} = -1, \\ v_4^{[0]} &= v_1^{[0]} + v_2^{[0]} = 2, \\ F_1^{[0]} &= y^{[0]} * v_4^{[0]} = -2, \\ F_2^{[0]} &= v_3^{[0]} * v_4^{[0]} = -2. \end{aligned}$$

Korzystając z tożsamości (6) *otrzymujemy* $x^{[1]} = F_1^{[0]} = -2$ *oraz* $y^{[1]} = F_2^{[0]} = -2$. *Propagacja pierwszych pochodnych przez graf wyrażenia* (5) *daje*

$$\begin{aligned} v_1^{[1]} &= 2 * x^{[0]} * x^{[1]} = -4, \\ v_2^{[1]} &= 2 * y^{[0]} * y^{[1]} = 4, \\ v_3^{[1]} &= -x^{[1]} = 2, \\ v_4^{[1]} &= v_1^{[1]} + v_2^{[1]} = 0, \\ F_1^{[1]} &= y^{[0]} * v_4^{[1]} + y^{[1]} * v_4^{[0]} = -4, \\ F_2^{[1]} &= v_3^{[0]} * v_4^{[1]} + v_3^{[1]} * v_4^{[0]} = 4. \end{aligned}$$

Ponownie korzystając z tożsamości (6) *otrzymujemy* $x^{[2]} = \frac{1}{2}F_1^{[1]} = -2$ *oraz* $y^{[2]} = \frac{1}{2}F_2^{[1]} = 2$. *Otrzymaliśmy początek rozwinięcia w szereg Taylora rozwiązania zagadnienia początkowego*

$$\begin{aligned} x(t) &= 1 - 2t - 2t^2 + \dots \\ y(t) &= -1 - 2t + 2t^2 + \dots \end{aligned}$$

bez wyznaczania pochodnych samego pola wektorowego. Dalsze iteracje pozwalają obliczyć współczynniki Taylora $x^{[n]}$, $y^{[n]}$ *dla dowolnie dużego, skończonego* n.

Ścisła metoda Taylora

Przykład 8 sugeruje, że można łatwo rozwijać w szereg Taylora rozwiązania zagadnień początkowych, o ile pole wektorowe jest funkcją prostą. Nie wiemy jednak, na jakim przedziale czasowym to rozwiązanie jest określone. Pokażemy teraz, że korzystając z arytmetyki przedziałowej oraz algorytmicznego różniczkowania można zdefiniować algorytm, który dowodzi istnienia rozwiązań dla *zbioru zagadnień początkowych* na jawnie określonym przedziale czasowym.

Wejściem algorytmu są:

- $\dot{x} = f(x)$ – równanie różniczkowe z prawą stroną będącą funkcją prostą,
- $[X]$ - **zbiór** warunków początkowych, najczęściej wektor przedziałowy,
- h – proponowany krok czasowy.

Algorytm oblicza

- $[Y]$ – taki, że $[X](0,h) \subset [Y]$ (tzw. **rough enclosure**)
- oszacowanie na $[X](h)$. W przypadku metody Taylora jest to

$$[X](h) \subset \sum_{k=0}^{r} X^{[k]}(0)h^k + [Y]^{[r+1]}(0)h^{r+1},$$

  dla pewnego $r > 0$,

lub zwraca **Failure**, jeżeli nie jest możliwe obliczenie $[Y]$ oraz $[X](h)$. Ten przypadek nie oznacza, że rozwiązania nie istnieją – po prostu algorytm nie był w stanie udowodnić, że są zdefiniowane na przedziale $[0,h]$.

Najważniejszym krokiem jest obliczenie rough enclosure. Podam tutaj najprostszy algorytm, tzw. First Order Enclosure. Poprawność tego algorytmu jest oparta o następujące proste twierdzenie.

Twierdzenie 4. *Niech $f\colon \mathbb{R}^n \to \mathbb{R}^n$ będzie gładkim polem wektorowym, $[X], [Y] \subset \mathbb{R}^n$ będą wektorami przedziałowymi. Ustalmy $h > 0$. Jeśli*

$$[X] + [0, h][f]([Y]) \subset \operatorname{int}([Y]),$$

to

- *dla $x_0 \in [X]$ rozwiązania zagadnienia początkowego (5) jest określone na przedziale $[0, h]$ oraz*
- *$x_0(t) \in [X] + [0, h][f]([Y])$ dla $t \in [0, h]$.*

Algorytm First Order Enclosure polega na zgadnięciu zbioru $[Y]$, a następnie sprawdzeniu, że spełnia on założenia Twierdzenia 4. Metod zgadywania może być oczywiście wiele. Jedną z najprostszych, ale zadziwiająco skutecznych predykcji jest określenie

$$[Y] = [X] + h * [-a, 1 + b][f]([X]),$$

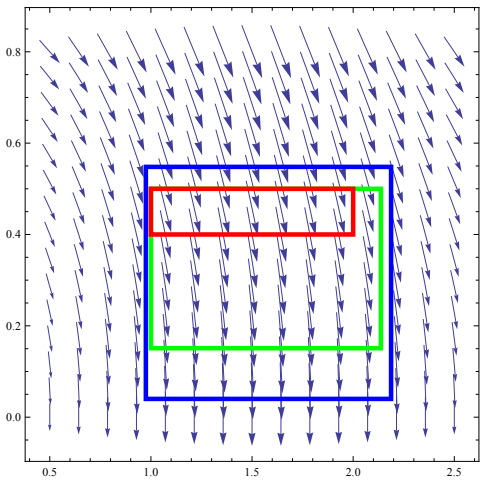
dla pewnych niedużych $a, b > 0$. Zilustrujemy to na przykładzie.

Przykład 9. *Rozważmy równanie $x'' = -\sin(x) + 0.1x'$. Ustalamy krok czasowy $h = 0.25$ oraz warunek początkowy $[X] = [1, 2] \times [0.4, 0.5]$. Obliczamy w arytmetyce przedziałowej*

$$[Y] = [X] + h[-.2, 1.5] * [f]([X]) \subset [0.9749, 2.1875] \times [0.04, 0.548],$$
$$[Z] = [X] + [0, h] * [f]([Y]) \subset [1.0, 2.137] \times [0.1502, 0.5] \subset \operatorname{int}([Y]),$$

co dowodzi, że $[Y]$ spełnia założenia Twierdzenia 4.

Zbiory $[X], [Y]$ i $[Z]$ zostały przedstawione na rysunku 9.



Rys. 9. Obliczenie rough enclosure dla danych z Przykładu 9: $[X]$ (czerwony) – zbiór warunków początkowych, $[Y]$ (niebieski) – obliczony kandydat na rough enclosure, $[Z]$ (zielony) – oszacowanie na trajektorie wychodzące z $[X]$ na przedziale czasowym $[0, h]$.

Podsumowanie

Artykuł stanowi bardzo krótkie wprowadzenie w tematykę komputerowo wspieranych dowodów w układach dynamicznych. Pominąłem wiele technicznych, często bardzo subtelnych szczegółów, bez których przedstawione metody są mniej efektywne lub wcale nie działają (nie da się otrzymać oszacowań lub są one zbyt duże, aby były użyteczne). Pominąłem również algorytmy obliczania oszacowań na odwzorowanie Poincarégo i jego pochodne – jest to temat na oddzielny artykuł.

Mam jednak nadzieję, że udało mi się zarysować ogólny schemat postępowania przy konstrukcji komputerowo wspieranego dowodu pewnej własności układu dynamicznego. Należy najpierw sprowadzić badany problem za pomocą dobrze dobranych abstrakcyjnych twierdzeń do skończenie wielu nierówności, które następnie mogą być sprawdzone za pomocą komputera.

Literatura

[1] CAPD – Computer Assisted Proofs in Dynamics group, `http://capd.ii.uj.edu.pl`.

[2] FADBAD++ – Flexible Automatic differentiation using templates and operator overloading in C++, `http://www.fadbad.com`.

[3] J.L. Lions, L. Lübeck, J.-L. Fauquembergue, G. Kahn, W. Kubbat, S. Levedag, L. Mazzini, D. Merle, C.O'Halloran, *Ariane 5 flight 501 failure report by the inquiry board*, `http://zoo.cs.yale.edu/classes/cs422/2010/bib/lions96ariane5.pdf`, dostęp 31 lipca 2019r.

[4] S.M. Rump, *Algorithms for Verified Inclusions - Theory and Practice*, In R.E. Moore, editor, Reliability in Computing, volume 19 of Perspectives in Computing, pages 109–126. Academic Press, 1988.

[5] A. Griewank, *ODE solving via automatic differentiation and rational prediction*, in Numerical Analysis 1995 (Dundee, 1995), Pitman Res. Notes Math. Ser. 344, Longman, Harlow, UK, 1996, pp. 36–56.

[6] *The IEEE Standard for Binary Floating-Point Arithmetics*, ANSI-IEEE Std 754, (1985).

[7] R.E. Moore, *Methods and Applications of Interval Analysis,* SIAM, Philadelphia, 1979

[8] O.E. Rössler, *An Equation for Continuous Chaos*, Physics Letters, Vol. 57A no 5, pp 397–398, 1976.

[9] O.E. Rössler, *An equation for hyperchaos*, Physics Letters A 71 , no.2-3, (1979), 155–157.

[10] I. Walawska and D. Wilczak, *An implicit algorithm for validated enclosures of the solutions to variational equations for ODEs*, App. Math. Comp., **291C**, (2016) 303–322.

[11] D. Wilczak, *Computer assisted proof of chaotic dynamics in the Rössler map*, Topological Methods in Nonlinear Analysis (2001) 1, Vol. 18, 183-190.

[12] P. Zgliczyński, *Computer assisted proof of chaos in the Hénon map and in the Rössler equations*. Nonlinearity, 1997, Vol. 10, No. 1, 243–252.