

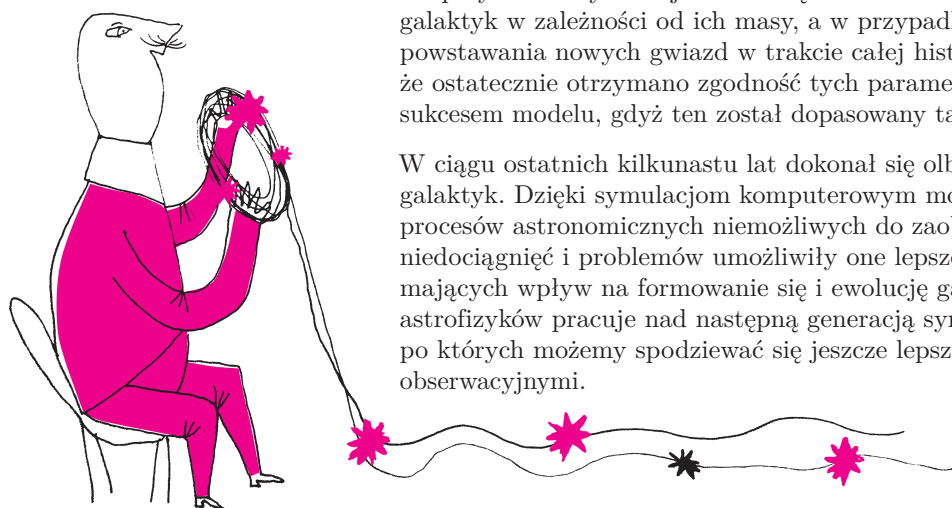
Illustris: www.illustris-project.org
EAGLE: icc.dur.ac.uk/Eagle/ oraz
eagle.strw.leidenuniv.nl/

Dzięki postępowi technologicznemu oraz dostatecznemu zrozumieniu istotnych procesów możliwe stało się wierne symulowanie reprezentatywnych kawałków Wszechświata. Dwie takie symulacje to Illustris oraz EAGLE. Na stronach internetowych obydwu projektów znajdują się modele oraz animacje z symulacji, przedstawiające m.in. powstawanie galaktyk eliptycznych oraz spiralnych. W obydwu projektach udało się odtworzyć wiele własności obserwowanych galaktyk, między innymi proporcje między galaktykami spiralnymi i eliptycznymi, skład chemiczny, wielkoskalowy rozkład neutralnego wodoru oraz rozkład galaktyk w gromadach.

Oczywiście, sukces symulacji poprzedniej generacji nie jest końcem poszukiwań, gdyż wciąż pozostaje wiele problemów do rozwiązania. W Illustris powstające galaktyki są nazbyt duże w porównaniu z rzeczywistymi rozmiarami. Innym problemem jest to, że gwiazdy w małych galaktykach powstawały zbyt szybko, więc ich końcowy wiek jest 2–3 razy większy od obserwowanego.

Najważniejszym problemem, który pozostaje do rozwiązania i który jest najbardziej niepokojący dla naukowców, jest problem wspomnianego już feedbacku. Niedostateczna rozdzielczość symulacji oraz niepełne zrozumienie odpowiednich procesów fizycznych skutkuje tym, że dostarczanie energii do ośrodka międzygwiazdowego przez gwiazdy i aktywne jądra galaktyk jest modelowane w sposób bardzo przybliżony, a nie wyprowadzane z podstawowych praw fizyki. Najkrócej mówiąc, dobierane są takie parametry, aby pewna własność populacji galaktyk jak najlepiej zgadzała się z rzeczywistością. Na przykład w symulacji EAGLE dążono do odtworzenia rozkładu ilości galaktyk w zależności od ich masy, a w przypadku Illustrisa – średniego tempa powstawania nowych gwiazd w trakcie całej historii Wszechświata. Zatem to, że ostatecznie otrzymano zgodność tych parametrów z obserwacjami, nie jest sukcesem modelu, gdyż ten został dopasowany tak, aby otrzymać taką zgodność.

W ciągu ostatnich kilkunastu lat dokonał się olbrzymi postęp w modelowaniu galaktyk. Dzięki symulacjom komputerowym możemy śledzić przebieg procesów astronomicznych niemożliwych do zaobserwowania. Mimo wszystkich niedociągnięć i problemów umożliwiły one lepsze zrozumienie procesów mających wpływ na formowanie się i ewolucję galaktyk. Obecnie kilka zespołów astrofizyków pracuje nad następną generacją symulacji kosmologicznych, po których możemy spodziewać się jeszcze lepszej zgodności z danymi obserwacyjnymi.



Równanie z dreszczykiem

*Piotr KRZYŻANOWSKI**

Jakiś czas temu Marek Bodnar z sąsiedniego Zakładu Biomatematyki pokazał mi niepozornie wyglądające równanie różniczkowe, które pojawiło się w pewnym modelu przebiegu choroby zakaźnej:

$$(1) \quad X'(\tau) = -\mu X(\tau) + \frac{1}{\epsilon} (\lambda X(\tau) (N e^{r\tau} - X(\tau)) - \gamma X(\tau)).$$

Liczby $\mu, \lambda, \epsilon, \gamma, r, N$ są stałymi, dodatnimi parametrami modelu (zob. [1, 2]). Niewiadomą jest funkcja $X(\tau)$ odpowiadająca liczbie chorych przypadających na jednostkę powierzchni. Zmienna niezależna τ to czas. Odpowiednio skalując zmienne występujące w równaniu, możemy sprowadzić je do prostszej, równoważnej postaci

$$(2) \quad x'(t) = x(t) \cdot (e^{At} - B - x(t)),$$

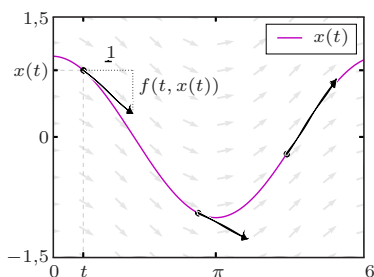
przy czym $A, B > 0$ zależą od parametrów oryginalnego zadania. Dodatkowo na rozwiązanie nakładamy warunek $x(0) = x_0$, gdzie $x_0 > 0$ jest kolejnym parametrem zadania.

*Instytut Matematyki Stosowanej i Mechaniki, Wydział Matematyki, Informatyki i Mechaniki, Uniwersytet Warszawski

[1] J. Banasiak, E.K. Phongi, *Canard-Type Solutions in Epidemiological Models*, [dx.doi.org/10.3934/proc.2015.0085](https://doi.org/10.3934/proc.2015.0085)

[2] J. Banasiak, *Bifurkacje dynamiczne i osobliwie zaburzone układy równań*, imsm.mimuw.edu.pl/konf30/slides/j-banasiak.pdf

Parametr x_0 odpowiada liczbie zarażonych w chwili $t = 0$.



Rys. 1. Wykres rozwiązania $x(t) = \cos(t)$ równania $x'(t) = -\sin(t)$ w każdym swoim punkcie (t, x) jest styczny do wektora $[1, -\sin(t)]$.

Jest to **równanie różniczkowe Bernoulliego** drugiego stopnia. Czytelnikom nie w pełni zaznajomionym z teorią równań różniczkowych wyjaśnimy, że wykres poszukiwanej przez nas funkcji $x(t)$, spełniającej równanie różniczkowe zapisane w ogólnej postaci

$$(3) \quad x'(t) = f(t, x(t)),$$

ma taką własność, że jest on styczny w każdym swoim punkcie $(t, x(t))$ do wektora o kierunku $[1, f(t, x(t))]$, zob. przykład na rysunku 1 dla $f(t, x) = -\sin(t)$. W przypadku równania (2), oczywiście, $f(t, x)$ jest inne, równe $x \cdot (e^{At} - B - x)$.

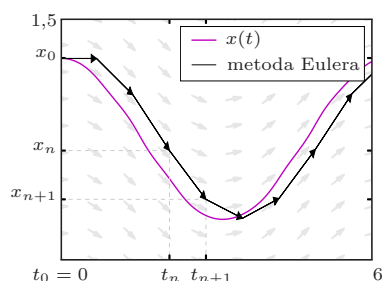
Równanie różniczkowe? Ależ to proste!

W dzisiejszych czasach zdawać by się mogło, że nie musimy rozumieć, czym jest równanie różniczkowe – wystarczy tylko wiedzieć, z jakiej biblioteki numerycznej skorzystać na swoim laptopie... Rzeczywiście: jedna z najlepszych (LSODE) jest wbudowana w darmowy pakiet obliczeń komputerowych GNU Octave; inne, równie znakomite, są używane przez MATLAB-a, Scipy, Mathematicę, itd. Dlatego codziennie naukowcy i inżynierowie z całego świata wykorzystują je do rutynowego numerycznego rozwiązywania równań różniczkowych na komputerze, ufając ich technicznej doskonałości i wbudowanej wiedzy, obejmującej kilkanaście (lub więcej) lat użytkowania w najróżniejszych warunkach.

Jak działa taka biblioteka? Najprostszą strategią numerycznej aproksymacji rozwiązania równania (3) jest przyjęcie, że na krótkich odcinkach czasu długości h funkcja f jest stała (innymi słowy, że rozwiązanie jest funkcją liniową), co prowadzi do tzw. **metody Eulera**:

$$x_{n+1} = x_n + h \cdot f(t_n, x_n),$$

gdzie $t_n = nh$ oraz (można udowodnić, że) $x_n \approx x(t_n)$ (zob. rys. 2). Nietrudno zgadnąć, że oprócz tej prościutkiej metody są też bardziej zaawansowane, np. schematy wielokrokowe lub metody Rungego–Kutty – również w wersjach, które dodatkowo automatycznie dopasowują długość h do rzeczywistego przebiegu rozwiązania tak, by zagwarantować spełnienie określonych przez użytkownika kryteriów tolerancji. Wspomniany powyżej LSODE używa właśnie schematów wielokrokowych z adaptacją długości kroku.



Rys. 2. Metoda Eulera, wymyślona przez króla matematyków podobno na potrzeby przybliżonego rozwiązania zadania z hydrauliki fontann dla króla Prus. Tutaj zastosowana do zadania z rysunku 1, dla $h = 0,7$.

Febra wykresu: rozpoznanie, diagnoza i kuracja

Używając wbudowanej w pakiet Octave biblioteki LSODE do rozwiązania (2) dla parametrów $B = 157,143$, $x_0 = 0,2$ i kilku wybranych wartości A , odpowiednich dla modelowanego zjawiska, dostaniemy ładne wykresy, takie jak na rysunku 3 na górze.

Jak zauważył Marek Bodnar (i dlatego pokazał mi to równanie, bo wie, że lubię takie smaczki), gdy *ustalimy* zestaw parametrów zadania, ale za to będziemy bawić się tolerancjami pracy LSODE, otrzymamy podobne, lecz jednak zdecydowanie *różne* rozwiązania (rys. 3 na dole). (Zobacz też uwagi J. Banasiaka w [2, str. 36]. Analogiczne zjawisko pojawi się także w MATLAB-ie i in.)

Jak to rozsądzić? Kto przeszedł kurs *Równań różniczkowych zwyczajnych*, ten zauważy, że (2) tylko troszkę różni się od klasycznego **równania logistycznego**

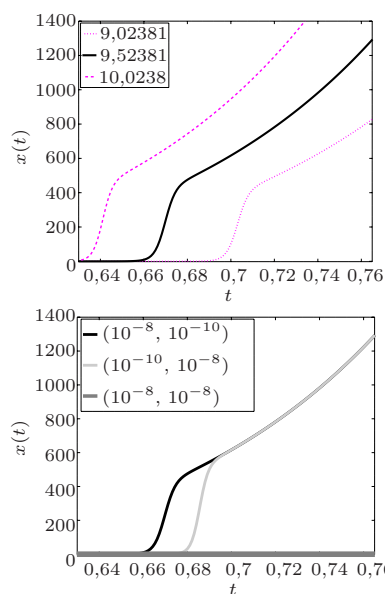
$$y'(t) = y(t) \cdot (C - y(t)),$$

dla którego *znamy jawny wzór* na rozwiązanie:

$$y(t) = \frac{C y(t_0)}{y(t_0) + (C - y(t_0)) e^{-C(t-t_0)}}.$$

Gdyby więc – podobnie jak w metodzie Eulera – przyjąć chwilowo, że człon e^{At} jest stałą wynoszącą, powiedzmy, e^{At_0} , to wtedy (2) zamienia się w równanie logistyczne z $C = e^{At_0} - B$.

Ta wiedza pozwoli nam rozstrzygnąć, które z uzyskanych przez nas rozwiązań jest akceptowalne. Wykorzystamy następujące twierdzenie porównawcze:



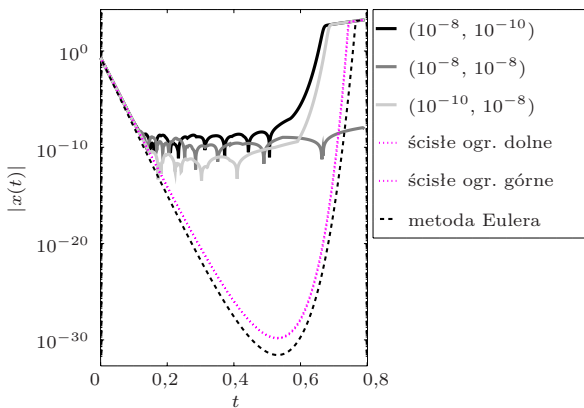
Rys. 3. U góry: rozwiązania dla różnych wartości parametru A modelu (2) tylko z pozoru wyglądają sensownie. Na dole: rozwiązania wyznaczone dla tej samej wartości $A = 9,52381$, ale przy różnych tolerancjach (ATOL, RTOL) pracy LSODE znacznie się różnią (jedno wygląda wręcz na zerowe), co jest bardzo podejrzane.

Twierdzenie (Peano). Jeśli funkcje $x, y : [0, T] \rightarrow R$ są różniczkowalne i spełniają dla $0 \leq t \leq T$ warunki:

$$x'(t) = f(t, x(t)), \quad y'(t) < f(t, y(t)), \quad y(0) \leq x(0),$$

to $y(t) \leq x(t)$ dla $0 \leq t \leq T$.

Z tego, co powiedzieliśmy wcześniej, wynika, że jako y wystarczy wziąć rozwiązanie równania logistycznego z C ciut mniejszym niż $e^{At_0} - B$, gdyż w naszym zadaniu $x(t)$ jest zawsze dodatnie. Łatwo też uzupełnić treść powyższego twierdzenia tak, aby otrzymać analogiczne dolne oszacowanie $x(t)$. Ponadto nic nie stoi na przeszkodzie, byśmy zastosowali je – podobnie jak w metodzie Eulera – nie od razu dla wszystkich t , ale wielokrotnie na krótkich odcinkach długości, powiedzmy, $H = 1$. W ten sposób dostajemy dwie – jak się okazuje, całkiem bliskie – krzywe, będące ścisłymi oszacowaniami z góry i z dołu dla prawdziwego (wciąż nieznanego!) rozwiązania (2): obie zostały naszkicowane na rysunku 4.



Rys. 4. Wykresy rozwiązań LSODE w skali logarytmicznej ukazują podejrzaną zachowanie dla małych wartości $x(t)$. Górne i dolne oszacowanie na bazie twierdzenia Peano są tak bliskie, że na rysunku wyglądają jak jedna krzywa.

Podobnemu wykresowi, z MATLAB-a, przyglądaliśmy się w poprzednim numerze *Delt*y.

Mt 20,16 Niestety, na rysunku 4 widzimy także, iż żaden wcześniej otrzymany wykres LSODE nie mieści się w wyznaczonym przez nas korytarzu, w którym musi przebiegać rozwiązanie. I, niczym w biblijnej przypowieści, prymitywna metoda Eulera ostatecznie wygrywa z wyrafinowanym LSODE: rozwiązania wyznaczone przez jedną z najlepszych numerycznych bibliotek okazują się bowiem dramatycznie kiepskie tam, gdzie wartości są bardzo małe: wykres trzęsie się jak w febrze, choć z teorii wiadomo, że powinien być gładki.

Patrząc na rysunek 4, zaczynamy rozumieć, o co chodzi. Schemat taki, jak LSODE ma wbudowany mechanizm sterowania długością kroku (w zamyśle służący *polepszeniu* jakości rozwiązania!), który jednak najwyraźniej gubi się, gdy rozwiązanie staje się w specyficzny sposób czułe na niewielkie odchylenia od dokładnej wartości, a tak dzieje się, gdy rozwiązanie przybliży się do zera.

LSODE (w końcu) też daje radę! W takiej sytuacji – gdy zadanie staje się zbyt trudne dla używanej metody – numeryk często proponuje: *Może więc zmieńmy... zadanie?!* Faktycznie, wprowadzając w (2) nową niewiadomą $y = \log x$, dostaniemy kolejne równoważne równanie:

$$(4) \quad y'(t) = -G + e^{Ft} - e^{y(t)}, \quad y(0) = \log(x_0),$$

którego rozwiązania mają znacznie spokojniejszy przebieg niż (2). LSODE rozwiązuje je bez trudu, więc kładąc następnie $x(t) = e^{y(t)}$, wyznaczamy znakomity wykres, nareszcie idealnie wpasowujący się pomiędzy górne i dolne ograniczenie. Nie pokazujemy tego wykresu, bo i tak w używanej przez nas skali będzie nieodróżnialny od krzywych ograniczających z rysunku 4.

Walka nie jest jeszcze wygрана

Ale w życiu takie rzeczy nie zdarzają się, prawda? – chciałoby się zapytać, mając w pamięci, że cała technologiczna nowoczesność wokół nas była zapewne wcześniej testowana w modelach komputerowych, które – jak przekonuje nas codzienne doświadczenie – najwyraźniej musiały *dać się* numerycznie rozwiązać w wystarczająco dokładny sposób. Generalnie, metody numeryczne rzeczywiście działają zdumiewająco dobrze; jednak trafiają się spektakularne niepowodzenia – takie, jak zatonięcie 26 lat temu pływającej platformy wiertniczej Sleipner (zakończone wstrząsem tektonicznym o mocy 3 stopni w skali Richtera i stratą rzędu 700 mln dolarów) – spowodowane tylko jednym czynnikiem: zlekceważeniem naturalnych ograniczeń standardowych pakietów obliczeniowych.

Warto o tym pamiętać, naciskając klawisz Enter... i nie tylko.

Parametry F, G w równaniu (4) zależą, oczywiście, od A i B .

O innych perypetiach wykorzystania komputera do wykonania obliczeń pisaliśmy m.in. w $\Delta_{09}^3, \Delta_{09}^6, \Delta_{09}^{10}$.

Ukuto nawet specjalny termin: *Computer-Aided Catastrophe*.