Introducing Dynamical Systems and Chaos Early in Computer Science and Software Engineering Education Can Help Advance Theory and Practice of Software Development and Computing

David Harel(✉) and Assaf Marron

Weizmann Institute of Science, Rehovot, Israel
david.harel@weizmann.ac.il

Abstract. Dynamical systems, i.e., systems that progress along time according to fixed rules, exhibit many special phenomena like the emergence of interesting patterns, bifurcation of behavior, the appearance of chaos despite determinism and boundedness, and sensitive dependence on initial conditions. Such phenomena are encountered in diverse fields, such as fluid dynamics, biological population analysis and economic and financial operations. The study of dynamical systems, their properties, and the mathematical and computerized tools for dealing with them, are often designated as part of advanced curricula in physics or mathematics. Consequently, many computer science students, perhaps the majority thereof, graduate without ever being exposed to such concepts. We argue that with the pervasiveness of dynamical systems and manifestation of their properties in the real world, these concepts should be introduced early on; in undergraduate studies in computer science and related fields, and perhaps even in high school. Available introductory courses demonstrate that only a minimal foundation of knowledge in mathematics is needed for the basic understanding of the key ideas. Such an introduction would deepen one's understanding of the world and highlight important capabilities and limitations of mathematical and software tools for analysis, simulation, testing and verification of complex systems. In turn, this can lead to enhancement and enrichment of languages, tools and methodologies for dealing with dynamical systems, and of research in computer science and software engineering in general.

1 Motivation

The concept of emergent entities and emergent properties is central to the study of complex systems. Examples include a traffic jam, a spinning tornado, a swarm of bees, the organization and behavior of an ant colony, and the partition of the unfathomable number of organisms in nature into millions of distinct species. Depictions of an ant eater befriending an ant colony in Hofstadter's seminal book Gödel, Escher, Bach [14] are particularly illuminating of the distinction between

T. Margaria and B. Steffen (Eds.): ISoLA 2021, LNCS 13036, pp. 322–334, 2021.
https://doi.org/10.1007/978-3-030-89159-6_20

the emergent entity, i.e., the friendly colony, and its constituent components, namely the poor ants who may be served as food to maintain this friendship with the anteater.

In our own research on biological evolution and of biological modeling [4,5], we have encountered extensive interest by scientists in the emergence of new patterns and order either from seemingly disordered behavior or, more often, from well specified and seemingly constrained local behavior [15,22,28]. This, in turn, led us more deeply into the realm of dynamical systems and chaos, in which concepts like emergence, bifurcation, sensitivity to initial conditions, fractals, and never-exactly-repeated behavior (which are explained briefly in Sect. 2) are dealt with thoroughly using analytical and computational tools [8,11].

This, combined with our interest in computer science (CS) education and in making deep CS concepts accessible to the general public (see, e.g., [13]), made us realize that ideas and notions associated with dynamical systems, chaos and emergence are often absent from the curricula of basic CS and software and system engineering (SE). A brief check of curricula in leading universities confirms this observation. And while dynamical systems and chaos (DS&C) are usually introduced as part of the disciplines of physics or mathematics, they are often considered to be advanced optional material and are offered as part of an elective or on the graduate level.

In this paper, we argue that concepts in, or properties of, DS&C are relevant to CS and software and system engineering students and professionals, and indeed to people from other areas who are interested in science and the observation of nature, and in engineering and philosophy. Thus, we claim, these concepts should be introduced early on, in undergraduate studies and perhaps even in high school. The scope of such an introduction can range from a single overview lecture, through a unit in a broader course, to an entire introductory course.

Some courses, books, lectures and papers already address the aim of making dynamical systems accessible to individuals with only basic mathematical background. For example, Devaney's introductory book [6] is targeted at undergraduate students who have had only a one year calculus course, and does not require knowledge of differential equations. Feldman's book [8] and his highly accessible series of videos from the Santa Fe Institute and the College of the Atlantic under the Complexity Explorer series [7] can also serve as excellent starting points for students and teachers new to this domain; they do not even require calculus and the relevant aspects of derivatives and partial differential equations are taught as an integral part of the DS&C course. Further support for the claim that these seemingly advanced concepts can be understood by students with a more limited mathematical background is provided by the research on teaching dynamical system to high school students [10]. An invaluable, and even less technical introduction to the history and the beauty of the field is offered in Gleick's book [11].

The goals of such introductory information include: (i) broadening one's perspective of the world in action, in line with the maxim attributed to Albert Einstein *"look deep into nature, and then you will understand everything better"*; (ii) laying a foundation for individuals who will later actually work with common dynamical systems in academia and industry; (iii) alerting professionals to the existence of dynamical-system and chaotic traits and unpredictable behavior in systems that might not be considered as such at first; (iv) highlighting requirements and gaps in techniques for development, analysis, simulation, testing and verification of complex dynamical systems, thus offering leads into research on programming languages and abstractions, algorithms, and theory; and, (v) making available to students and scientists the theory and tools developed for dynamical systems that have been shown to be of value when working on problems that are considered to be at the core of computer science.

2 The Subject Matter: Key Phenomena in Dynamical Systems and Chaos

A dynamical system is defined as a mathematical system that progresses in time according to fixed rules. These rules can operate in discrete time via iterative recurrence relations, of the form $x_{n+1} = f(x_n)$, where x_n and x_{n+1} are the states of the system at time n and $n+1$, respectively, or continuously, expressed as differential equations like $\frac{\partial x}{\partial t} = f(x, y, z)$, $\frac{\partial y}{\partial t} = f(x, y, z)$, and $\frac{\partial z}{\partial t} = f(x, y, z)$, where $\langle x, y, z \rangle$ represents the state of the system (in this case, in a three dimensional space) and the derivatives are rates of change with respect to time of each component of the state. In both the discrete and continuous cases, the next state, or the next change in state, are a function of only the current state. All seemingly external events and conditions are incorporated as internal parameters of the model, and the only role of the passage of time has to do with the step size in calculating the next system state. The absolute wall-clock time, or the time that has elapsed since the beginning of a simulation or an observation, are not essential parts of the model.

In dynamical systems, the functions depicting changes in state are often nonlinear, that is, the changes in output are not directly proportional to the changes in input. This non-linearity makes the mathematical analysis more difficult, which may be one of the reasons for why the topic is not taught earlier on. However, as in the quote attributed to Stanislaw Ulam [1]: *"Using a term like nonlinear science is like referring to the bulk of zoology as the study of non-elephant animals."*, the pervasiveness of nonlinear systems mandates that scientists, students and engineers should be more familiar with them.

Specifically, the main phenomena we deem relevant to this paper's proposal are:

1. **Emergence:** General definitions of emergent entities and properties often relate to properties of a system, or a set of entities, that are not expressed by any one of the components of the system or elements of the set (see, e.g., [4] and references therein). In dynamical systems, one often observes unexpected patterns, arrangements, and self organization. Salt crystals, ocean waves, convection rolls in a slowly-heating liquid, flocks of flying birds, and living cells are examples of such emergent entities. More specifically, consider an autonomous vehicle (AV) out on the field, at some large distance from

a designated tree. The AV is programmed to always proceed in a direction that is close to being perpendicular to its line of sight to that tree, and to control the direction angle so as to keep the distance more or less constant. While these instructions are local, external observers would readily see that the vehicle's trajectory/orbit/phase-space forms a circle; also, local observers, like additional AV instrumentation with memory, or a human who is present in the AV, can also notice the circular nature of the trajectory.

2. **Bifurcation:** Some dynamical systems exhibit great qualitative differences in observed behavior when certain parameters are subjected to minute changes. For example, consider the famous logistic map function $x_{n+1} = x_n \cdot r \cdot (1 - x_n)$ (see, e.g., [6], and Fig. 1). It can be viewed as a population x, where $0 < x < 1$ (representing, say, the portion of the area of a Petri dish covered by bacteria) grows by a factor of r every time unit, and is restrained by a factor of $1-x$. The trajectory is computed by starting at some arbitrary initial value $0 < x_0 < 1$, and iteratively computing the map. For all values of r, with $r < 3$, each of the trajectories formed by the iterative operation of the map converges to a single value (with different fixed points for different values of r); for $3 < r <\sim 3.44$, each of the different maps eventually oscillates between two values; for $\sim 3.45 < r <\sim 3.54$ the maps long-term behavior is an oscillation with a period of 4, etc.; and, for $r = 4$ and other values, the function behavior is chaotic, visiting in an unpredictable order the entire $[0, 1]$ range.

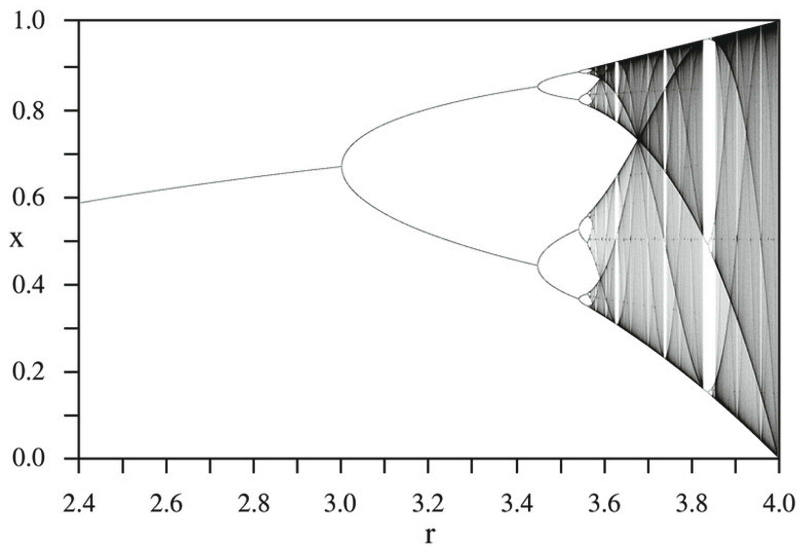

Fig. 1. The logistic map $x_{n+1} = x_n \cdot r \cdot (1 - x_n)$. The graph shows, for each value of r, the values that the map converges to after many iterations. For $r = 2.6$, this is a single value $\sim$0.61; for $r = 3.2$, the map oscillates between two values, $\sim$0.8 and $\sim$0.51; for $r = 3.5$ the oscillation period is 4: $\sim$0.50, $\sim$0.87, $\sim$0.38, $\sim$0.83; and for $r = 4$, the map yields chaotic coverage of the entire range between 0 and 1. See the body of the text in the bifurcation paragraph for more details. Image source: Wikipedia; under fair-use licensing.

The transitions between these behaviors are sudden and occur in very narrow ranges of values of r. Similar sudden changes in behavior occur in other dynamical systems like the changes in patterns and periods in the behavior patterns of a dripping faucet in response to the incoming water flow, or the number of convection rolls in a heated container as a function of the rate of change in temperature.

3. **Sensitivity to initial conditions:** This phenomenon, highlighted by the discoveries of Lorenz (see Fig. 3), and which is often termed "the butterfly effect", means that certain functions, even very simple ones, can produce very different system trajectories when starting at arbitrarily close, yet distinct, states. These distinct states commonly reflect not a particular choice, or an uninvited deviation from some desired reality, but merely a measurement error due to instrument limitations, or constraints imposed by the finite representations of numbers in computing. The term butterfly effect alludes to the difficulty in predicting the long term path of a storm, when the formulas in the model are sensitive to minute details, such as, figuratively speaking, whether a far-away butterfly, which might have been included in the model, did or did not flap its wings at a certain point in time.
4. **Unpredictable chaotic behavior:** In the study of dynamical nonlinear systems, behavior is considered to manifest chaos, or be chaotic, when it is unpredictable, but not because it depends on randomness or pseudo randomness; the system's state changes, and its infinite trajectory through all its possible states, are governed by deterministic mathematical rules. However, these rules are such that (i) they never yield the same exact state twice, despite being bounded within a closed region of $\mathcal{R}^n$, and (ii) they are sensitive to initial conditions. The non-repetition of states within a bounded region causes the behavior to be non-periodic and to require ever-growing precision in the representation of real numbers (in order to distinguish near states). The non-periodicity, the sensitivity of the rules to initial conditions and the inevitable finite precision of any computing facility, then contribute to making it virtually impossible to predict the state that the system will be in beyond some near-term horizon. This apparent contradiction between determinism in intended system behavior, which is the very essence of programming, algorithms and computation, and the appearance of long-term behavior as a seemingly disordered random mess may be settled when considering a highly tangled thin wire or fishing line, as in Fig. 2. Clearly there is no one analytic formula that can tell us where in space each molecule or each segment of the fishing line resides, based on its distance from one of the ends of the line. Still, if an ant were to walk the length of this line, starting at one of its ends, the ant's near-term general direction would be reasonably well defined for any location on the line. It is also intuitive to think that if the fishing line can be infinitesimally thin, one can always insert (indeed, thread) an additional length of it into the tangle without disrupting or cutting through existing line segments. For a more formal example, we return to the logistic map mentioned earlier, and consider $r = 4$, where the behavior is chaotic. Given a fully specified initial value of x, if one wishes to compute

the map value after n iterations, one must first compute all preceding $n-1$ iterations to unbounded precision. One may then ask whether or not such a process can be considered to be a prediction.

Additional phenomena manifested by DS&C include the following: the formation of fractal structures, i.e., self-similar recursive structures and behaviors; the existence of strange attractors, i.e., the convergence of behavior towards a chaotic, sometimes fractal pattern (such as the three dimensional figure-eight continuous trajectory of Lorenz equations system shown in Fig. 3, or the fractal boomerang shape formed by the Hènon Map shown in Fig. 4, that is formed over time by points being drawn one at a time, in different locations); and, the existence of universal mathematical parameters, like Feigenbaum's constant(s), which appear in highly disparate systems, ranging from the logistic map and other quadratic maps, through a dripping faucet, to the formation of convection rolls. See, e.g., [9] for more details.

Fig. 2. A single filament fishing line. The "path" of the thread illustrates unpredictable, yet deterministic and bounded behavior, in which any particular $\langle x, y, z\rangle$ coordinate in space is visited at most once; the path is very sensitive to the precision of measurement and calculation; any misstep by someone following the path of the thread could cause a transition to a different segment of the thread resulting in a very different trajectory. For an infinitely thin thread, new paths can always be found without physically intersecting, i.e., sharing an absolute coordinate, with an already traversed location. Image source: Wikipedia; under fair-use licensing.

3 Linking DS&C Tenets to General Education and to CS and SE

In this section we offer several perspectives on how DS&C is tied to classical computer science and software development, and why learning DS&C early may be beneficial to scientists and engineers.

3.1 DS&C Phenomena are Real and Pervasive

Most generally, the phenomena discussed in the previous section are real, both in nature and in their abstract mathematical manifestations, and there is a general consensus about them being aesthetic in some sense. Hence there may not be a need for further justification for including them at some level in general education and in scientific and engineering curricula, and for having CS and SE be part of the disciplines offering languages, tools, methodologies and theoretical foundations for dealing with these phenomena.

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \sigma(y - x)$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = x(\rho - z) - y$$

$$\frac{\mathrm{d}z}{\mathrm{d}t} = xy - \beta z$$

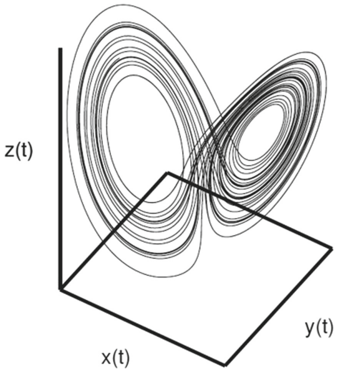

Fig. 3. Lorenz equations and trajectory. For any location of "the tip of the pen" drawing this graph in a three dimensional space, its speed along each of the axes, and thus its direction, is given by the respective partial derivative equations on the left. The result is the graph on the right. The equations are simple, and involve only elementary arithmetic operations over the current coordinates and some constants. Regardless of the starting point, the orbit is persistently attracted towards this well recognized, emergent, three dimensional "figure eight" shape. Still, given any current position of the traveling pen, it is impossible to predict far into the future in which lobe of the figure eight the pen will be at any particular time. Images sources: [7], under fair use licensing

3.2 CS and SE Already Deal with DS&C

At the other extreme of the links between DS&C and CS and SE is the fact that many sub-fields of CS and SE, as well as computer applications in other areas, already deal directly with dynamical systems. These include fluid/airflow dynamics, weather prediction, we transportation, robotics, manufacturing automation, economics and finance, storage/warehousing planning, disease control, livestock and agriculture management, and a variety of modeling and analysis applications in the study of biology, chemistry and physics. Therefore, it is only natural to expect CS and SE to be prepared to step in, attempting to address any technological, methodological or theoretical gaps or needs that may arise in such projects.

3.3 DS&C Issues May Emerge in *any* System

In between the two extremes listed in the preceding two subsections, we now list considerations related to DS&C concepts that affect requirements in the design of *any* system, whether perceived as a dynamic one or not. The bibliographic references given for each item serve to illustrate the issues via examples of in-depth treatment of well-scoped problems with relevant algorithms. Formulating succinct methodological principles that engineers and scientists should bear in mind, is yet to be done.

Pattern emergence. The emergence of unexpected patterns in images and other sensor input may cause incorrect classification or identification. This

$$\begin{cases} x_{n+1} = 1 - ax_n^2 + y_n \\ y_{n+1} = bx_n \end{cases} \qquad a = 1.4 \quad b = 0.3$$

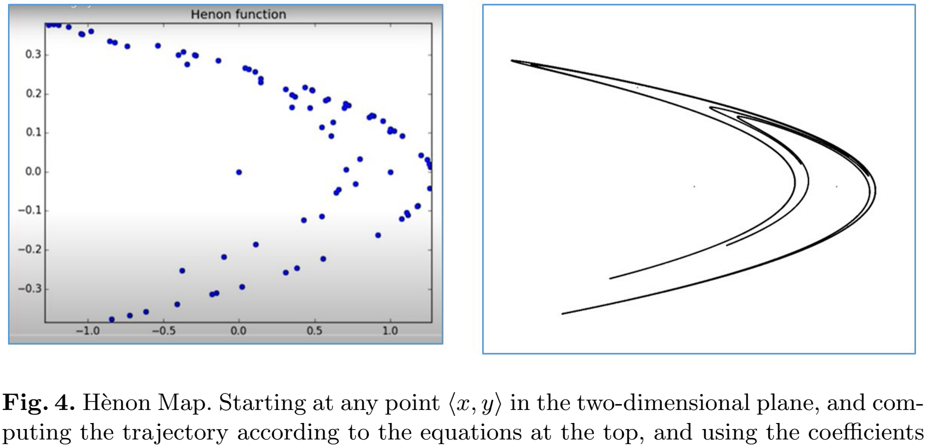

Fig. 4. Hènon Map. Starting at any point $\langle x, y \rangle$ in the two-dimensional plane, and computing the trajectory according to the equations at the top, and using the coefficients $a = 1.4, b = 0.3$, consecutive system states, i.e., coordinates of newly drawn points, may not be close to each other; after many steps, a boomerang-like shape emerges; the shape is fractal: when enlarging a sub-frame of the trajectory containing multiple lines (not shown here), one sees that each line is actually comprised of multiple thinner lines whose distances from each other are in proportions that are similar to the proportions of line distances in the original frame. Images sources: [9], under fair use licensing.

emphasizes the importance of redundancy in sensors of environment states and in the processing and analysis of their inputs (see, e.g., [2]) for causes of mistakes in image classification). Furthermore, such emergence of unexpected behavioral patterns may render a system more vulnerable to attacks that take advantage of the induced predictability, or cause excessive wear and tear in the system or in its environment (see, e.g., [18] for how repetition of a desired robot's path may form ruts in a field's soil).

Limitations in model-based prediction. Often, systems plan their behavior using on model-based algorithmic predictions. The development of such models and algorithms must accommodate the possibility of bifurcations in the behavior of the system and/or their environment throughout the allowable range of parameters (see, e.g., [12]). Furthermore, when an autonomous system explores new environments and new operational parameters, where it is not assured of the absence of bifurcations, it must be prepared for situations in which a very small change in parameters can cause dramatic and unexpected changes in behavior patterns.

Sensitivity to initial conditions. In computational/algorithmic models that affect operational decisions, the design must incorporate the level of sensitivity to initial conditions and to the finite precision in representing the current state, and set its prediction horizon accordingly. And, still, confident as the system may be in its predictions within the safe horizon, it must include mechanisms for appropriately reacting to unpredicted events and conditions, and, when possible, adapt its prediction process accordingly (see, e.g., [16,17]).

Chaotic behavior in classical algorithms. Chaotic behavior is observed in a variety of algorithmic contexts that do not originate in dynamical systems. One example is the so called "randomness" of digits in the number Pi (π) (more precisely, π is assumed to be *normal*, a property which refers to the observed uniform distribution of the appearance of digits and combinations thereof). Another example is the chaotic behavior of algorithms carrying out fast gradient descent in linear systems [24], where there is a non-monotone decrease in the norm of the residual vector.

3.4 CS and SE Research Can Help Close Gaps in Dealing with DS&C

There is a certain amount of published work on design principles for developing dynamical systems. For example [27] focuses on agent based models, [25] focuses on coding of society models, and [23] proposes principles for environmental and climate models. However, it appears that this sub-field is not as mature as software and system engineering for, say, traditional information processing systems, or control systems. Developing models of or controllers for dynamical systems can benefit from a variety of language idioms, and tools and methodologies, that are not readily available in existing languages and platforms for dealing with the various aspects of dynamical systems, such as the Modelica language (and tools implementing it), MATLAB and Simulink, NetLogo, AnyLogic, or Berkeley Madonna. To illustrate the need for language idioms and abstractions, we now list examples of challenges in monitoring a real or modeled dynamical traffic system. We focus on the detection, reaction to, and simulation of emergent entities (both expected and unexpected); languages for describing complex emergent behavior; dynamic incorporation of emergent entities as active programmable agents in a system, and more. Note that a human observer can readily handle the challenges in the examples below, yet programmed solutions require sophisticated procedural code:

- Detecting the existence of a traffic jam, and measuring its properties, such as its length. Note that the traffic jam entity exists despite its constituents being transient and dynamic, as vehicles leave its "head" and others join its "tail".
- Detecting a group of vehicles, like a truck convoy or a group of motorcyclists, and determining its relationship to other road users; for example, can others pass it safely?
- Detecting an unusual pattern in overall flow, as when a slow vehicle in a middle lane causes other vehicles to have to pass it on the right and left, with the difficulties and risks of changing lanes and merging.
- Detecting patterns in space and time; for example, whether the presence of certain kinds of vehicles, or of a certain kind of driver behavior is now more frequent/common than in the past.

While some such functions are carried out today using standard programming features, language idioms and development tools that address these directly could be of great value. And, not only can this be an important direction for computer scientists, we believe that studying DS&C can help attract undergraduate students to pursue graduate studies and research in this domain, further advancing the field.

3.5 Techniques Developed for DS&C Can Help Tackle Classical CS and SE Challenges

Approaches from dynamical system theory have been shown to be applicable in a variety of areas that are traditionally considered to be part of the computer science discipline. Example of such areas and one problem within each area are listed below. See the references for more details and examples.

- **Dynamical search** [19]**.** Consider the search for a minimum of a continuous function f in an interval where f is known to have only one such minimum; the search algorithm samples $f(x)$ at certain points within the current search interval e_n, and then narrows down the search interval $e_{n+1} = \psi(e_n)$ where ψ uses some algorithm-specific rules and the most recently found values of f.

This iterative computation of the intervals can be viewed as a dynamical system, and its behavior and convergence can then be analyzed using dynamical systems theory.

– **Algorithms for NP-hard problems** [21]**.** Consider the problem of partitioning a graph into equal size sets while minimizing the weights of cut edges; this problem arises in a range of settings, including gene networks, protein sequences, Internet routing algorithms and many more. To balance the cuts, the problem is often stated as minimizing the ratio between the inter-connection strength and the size of individual clusters. In this form, the problem is apparently intractable (NP-complete). An algorithm that uses dynamical system properties was developed for this task. It propagates waves in a graph in a completely decentralized setting, and has been shown to be orders of magnitude faster than existing approaches.
– **Machine Learning with Dynamical Systems** [3,20,26]**.** There is a growing body of work for introducing continuous dynamical system behavior and analysis into the theory of machine learning (ML) and neural nets. Elements of ML, like the propagation of information and computation within a neural net, the iterative training process that modifies the structure of a neural net based on prior results, and/or the behavior of the data itself in space (e.g., the processed image) are treated as continuous dynamical processes described using ordinary and partial differential equations.

4 Conclusion

Familiarity with the concepts and phenomena associated with nonlinear dynamical systems and chaos can enrich and enhance the work of scientists, engineers and educators in many fields. In particular, in the context of computer science and software & system engineering we expect that such broader and deeper awareness can trigger additional learning followed by valuable development and enhancement of languages, tools and methodologies. Based on already existing introductory material that does not require advanced knowledge of mathematics, we also believe that the first steps of such a shift in computer science education are in fact possible.

Acknowledgements. We thank Dror Fried for valuable comments throughout the development of this paper. Many thanks also to Irun Cohen, Smadar Szekely, Guy Frankel, and Meir Shani for constructive discussions and comments.

References

1. Campbell, D., Farmer, D., Crutchfield, J., Jen, E.: Experimental mathematics: the role of computation in nonlinear science. Commun. ACM **28**(4), 374–384 (1985)
2. Chakraborty, P., Adu-Gyamfi, Y.O., Poddar, S., Ahsani, V., Sharma, A., Sarkar, S.: Traffic congestion detection from camera images using deep convolution neural networks. Transp. Res. Rec. **2672**(45), 222–231 (2018)

3. Chen, R.T., Rubanova, Y., Bettencourt, J., Duvenaud, D.: Neural ordinary differential equations. arXiv preprint arXiv:1806.07366 (2018)
4. Cohen, I.R., Harel, D.: Explaining a complex living system: dynamics, multi-scaling and emergence. J. R. Soc. Interface **4**(13), 175–182 (2007)
5. Cohen, I.R., Marron, A.: The evolution of universal adaptations of life is driven by universal properties of matter: energy, entropy, and interaction. F1000Research **9** (2020)
6. Devaney, R.L.: A First Course in Chaotic Dynamical Systems: Theory and Experiment. Chapman and Hall/CRC Press, Boca Raton (2020)
7. Feldman, D.: Complexity Explorer Course on Dynamical Systems and Chaos. https://www.youtube.com/playlist?list=PLF0b3ThojznQwpDEClMZmHssMsuPnQxZT. Accessed May 2021
8. Feldman, D.: Chaos and Dynamical Systems. Princeton University Press, Princeton (2019)
9. Feldman, D.P.: Chaos and Fractals: An Elementary Introduction. Oxford University Press, Oxford (2012)
10. Forjan, M., Grubelnik, V.: How well do students in secondary school understand temporal development of dynamical systems? Eur. J. Sci. Math. Educ. **3**(2), 185–204 (2015)
11. Gleick, J.: Chaos: The Amazing Science of the Unpredictable. Vintage Publishing, New York (1998)
12. Guan, X., Cheng, R., Ge, H.: Bifurcation control of optimal velocity model through anticipated effect and response time-delay feedback methods. Phys. A: Stat. Mech. Appl. **574**, 125972 (2021)
13. Harel, D., Feldman, Y.A.: Algorithmics: the Spirit of Computing. Pearson Education, London (2004)
14. Hofstadter, D.R.: Gödel, Escher, Bach: An Eternal Braid; 20th Anniversary Edition. Penguin Books (1999)
15. Kauffman, S.A.: Investigations. Oxford University Press, Oxford (2000)
16. Kewlani, G., Crawford, J., Iagnemma, K.: A polynomial chaos approach to the analysis of vehicle dynamics under uncertainty. Veh. Syst. Dyn. **50**(5), 749–774 (2012)
17. Marron, A., Limonad, L., Pollack, S., Harel, D.: Expecting the unexpected: developing autonomous-system design principles for reacting to unpredicted events and conditions. In: Proceedings of the IEEE/ACM 15th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, pp. 167–173 (2020)
18. Ostafew, C.J., Schoellig, A.P., Barfoot, T.D., Collier, J.: Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking. J. Field Robot. **33**(1), 133–152 (2016)
19. Pronzato, L., Wynn, H.P., Zhigljavsky, A.A.: An introduction to dynamical search. In: Pardalos, P.M., Romeijn, H.E. (eds.) Handbook of Global Optimization, pp. 115–150. Springer, Boston (2002). https://doi.org/10.1007/978-1-4757-5362-2_4
20. Ruthotto, L., Haber, E.: Deep neural networks motivated by partial differential equations. J. Math Imaging Vis. **62**, 1–13 (2019)
21. Sahai, T.: Dynamical systems theory and algorithms for NP-hard problems. In: Junge, O., Schütze, O., Froyland, G., Ober-Blöbaum, S., Padberg-Gehle, K. (eds.) SON 2020. SSDC, vol. 304, pp. 183–206. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-51264-4_8
22. Schrodinger, E.: What is Life? The Physical Aspect of the Living Cell. Cambridge University Press (1951)

23. Simm, W.A., et al.: SE in ES: opportunities for software engineering and cloud computing in environmental science. In: Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Society, pp. 61–70 (2018)
24. van den Doel, K., Ascher, U.: The chaotic nature of faster gradient descent methods, pp. 1–27. The University of British Columbia, Canada (2011)
25. Vendome, C., Rao, D.M., Giabbanelli, P.J.: How do modelers code artificial societies? Investigating practices and quality of NetLogo codes from large repositories. In: 2020 Spring Simulation Conference (SpringSim), pp. 1–12. IEEE (2020)
26. Weinan, E.: A proposal on machine learning via dynamical systems. Commun. Math. Stat. **5**(1), 1–11 (2017)
27. Williams, R.A.: Lessons learned on development and application of agent-based models of complex dynamical systems. Simul. Model. Pract. Theory **83**, 201–212 (2018)
28. Wolf, Y.I., Katsnelson, M.I., Koonin, E.V.: Physical foundations of biological complexity. Proc. Natl. Acad. Sci. **115**(37), E8678–E8687 (2018)