

Exploring Discrete Dynamics : Book Review

David Burraston

Exploring Discrete Dynamics by Andrew Wuensche, published in 2011 by Luniver Press, 538 pages, 8×10in paperback.

Chance favours only the prepared mind – Louis Pasteur 1854

Discrete Dynamics Lab (DDLab) is a stunning piece of software for exploring cellular automata (CA) and other discrete dynamical networks, which are powerful tools in our quest to understand the complexities of life, nature and the many complex systems that intertwine our daily lives. As we forge ahead into the 21st Century and expand the science of complexity into the foundation of a sustainable future [1, 2, 3], we need tools for application, understanding and knowledge making. DDLab is one of the great tools that has helped to lay down some of these foundational concepts and will continue to do so as more people engage with this incredible software.

The DDLab manual, up until now, has been a purely on-line affair, freely downloadable as a PDF. However, over the last few years, Andrew Wuensche has been beaver away at the manual, now also available as a printed book. Having first printed out this comprehensive manual back in 2003 I have been loath to do a second printing and eagerly awaited the book publication, delving into the freely available online PDF updates as they became available. Now, at last, my luggage will be lighter, I have a large chunk of recyclable paper and a spare A4 ring binder! But, of course, that is only a very minor advantage compared to the invaluable contribution this new book will make to educationers, practitioners and researchers in this field.

Compiling this book (and the software) must have been a truly monumental task, and consequently Exploring Discrete Dynamics (EDD) is more than just an operations manual [4]. Contained within the book are many useful examples and a liberal sprinkling of high quality references which demonstrate the wide ranging areas encompassed by DDLab. In addition, Wuensche has an elegant way of introducing the underlying concepts simply and clearly, this is a rare thing for a “manual” to achieve despite many authors best intentions. Another impressive feature of EDD, which contributes to this clarity of exposition, is the vast number of well presented illustrations. A very useful Chapter 4 on “Quick Start Examples” enables both beginners and seasoned practitioners of complexity science to get hands on experience with the interface in a matter of minutes. The thoughtful inclusion of a glossary is also invaluable for a quick reference to many of the concepts in the book.

I will not give a detailed account of the DDLab software in this review, as this has been adequately covered elsewhere [5, 6]. Instead, I will outline some of the ways in which I have found DDLab useful in my generative music practice (with pointers to the book), and how it has enabled me to make my own discoveries regarding CA rule space structure [7, 8]. I had been using CA for many years in the 1990's to make music by writing my own software in order to generate musical patterns. After a while I felt I needed a better understanding of CA theory in order to make my own contribution to the scientific field from the perspective of an artist.

In the early 2000's when I first began using DDLab it was a result of reading Wuensche and Lesser's classic book on the global dynamics of CA [9]. This book introduced (among other things) the concept of attractor basins; state transition graphs representing the global behaviour of the system from all possible states as a field, containing one or more basins for a particular system size. I naturally began by

exploring basin of attraction fields and quickly realised that the atlas data in the W&L book was unavailable in electronic form. So as a first exercise I set myself the task of making the first (and at present only) definitive electronic version of the atlas using DDLab and the EDD book. Data files of basin field data were produced for 1D CA as part of Appendix E on the CD-ROM included with my PhD thesis [8,10]. Basin of attraction field data produced by DDLab can now be produced in a number of formats. The data for the original atlas also exists in a scanned format only, from the original printouts. During the process of creating the digital atlas I learned about the many different options for presenting basin field data, as well as how to export this data to text files. See Chapter's 23 to 30 in EDD, and the general section on filing in Chapter 35. The states can now be produced in hexadecimal, a more compact representation than binary, which was originally used (although binary is still an option). The benefit of having the data in this form is the ease of accessing and printing out specific basin information, and for further research. Computing this data was a great experience for learning DDLab and getting around the manual. Additionally, I also rendered the complex 5 neighbour rules in Wuensche's PhD thesis [10, 11]. All these rules have been computed, with system sizes ranging from 1 cell to between 24 and 31 cells, where practical. In all cases at least the minimum system size from the original atlas have been computed.

Producing attractor basin images was the next area I approached, and this is something that is really enjoyable. There are plenty of different layout options to customize how images can be presented and saved. For example, nodes in the basin can be labelled in binary, decimal or hex, or even left unlabelled for those really large basins. Branching angles and basin orientation can be specified. Subtrees can be created from random or specific seeds, and the seed state can also be saved to a separate file. Attractor basin layout can be manipulated to create really clear and informative images. Another aspect that I found extremely useful was the state space matrix plot, where the left and right halves of the states are plotted against each other. In addition to being very useful in my research, the visual patterns they produce are astoundingly beautiful.

Once I had gained more experience with DDLab I decided it was time to begin investigating something that had been troubling me for some time. In the late 1990's I had begun some experiments to build a hardware CA module for electronic music. I chose rule 90 as a test rule, because it produced the classic Sierpinski Triangle and, well, you've got to pick somewhere to start so I chose a bit of chaos. I decided on 8 cells as the system size, not realising back then this results in an evolution to all zeros! Initially I thought my design was to blame, and became quite intrigued when I could not find any errors. I later realised that it wasn't an error on my part, but exactly the right result. This intrigued me greatly: Why did a chaotic rule evolve to a totally ordered state in this instance? I decided that DDLab was the perfect tool for investigating this question. Strangely enough, it ended up providing me with a unique approach to a key problem: the structure of rule space.

I began by setting up DDLab to render the 8 cell attractor basin for rule 90 as a series of subtrees and labelled the nodes in decimal. I was now in for an even bigger surprise. The decimal numbers on the nodes were suddenly very familiar, they had a similarity with the way the elementary rules are clustered in complimentary transforms. Something interesting was going on here, so I printed all the subtrees out and decided to investigate further. Much to my surprise this state space organisation, **when considered as a representation of rule space**, resulted in an interesting grouping of rules as well as a navigable graph structure. I annotated the subtree printouts by hand (see Fig 1 for an example page), cross referencing them in more detail with the rule clusters in Wuensche & Lesser's book [9]. I had discovered that

CA could self-organise their own rule space. Further detail on these results can be found in my PhD thesis [8]. This investigation and discovery was made much easier by using DDLab and the EDD book.

All aspects of DDLab covered in the EDD book, and many interesting years of discovery lie ahead. There are many other ways of viewing dynamics, creating your own networks, analysing the resultant dynamics, visualising the network layout, classic parameters such as Z and λ , producing comprehensive data files, networks with multiple states, etc.

DDLab software (including the source code) and related publications are freely available online from Wuensche's web site [12]. If you're wondering whether to print out the book or buy a copy, my advice is to have the PDF on your computer and get a copy of the book to keep handy. For those wishing to get involved in the fascinating world of CA and discrete dynamic networks, a copy of DDLab running on your computer and this book by your side will be an invaluable tool. If you have as much fun with DDLab and this book as I have, you will be in for truly rewarding experience.

References

- [1] Burraston, D. (2012) Creativity + Complexity = Win Win, *Leonardo Electronic Almanac*. MIT Press (Invited essay)
- [2] J. Norberg and G. Cumming, *Complexity Theory for a Sustainable Future*, (Columbia University Press, 2008).
- [3] Mitchell, M. (2009). *Complexity: A Guided Tour*, Oxford University Press
- [4] Wuensche, A. (2011) *Exploring Discrete Dynamics*, Luniver Press.
- [5] Adamatzky, A, (1999) Discrete Dynamics Lab: 1992-1999, *Kybernetes*, Vol.28, No.8 and 9
- [6] Myers, J. E. (1997) Discrete dynamics lab (DDLab), *Complexity*, Vol.3, No.1.
- [7] Burraston, D. (2007) Fundamental Insights on Complex Systems arising from Generative Arts Practice. *Leonardo* Vol 40 (4), MIT Press.
- [8] Burraston, D. (2006) *Generative Music and Cellular Automata*. PhD Thesis, Univ. Technology Sydney, Australia. <http://noyzelab.com/research/research.html>
- [9] Wuensche, A., and Lesser, M.J. (1992), "*The Global Dynamics of Cellular Automata; An Atlas of Basin of Attraction Fields of One-Dimensional Cellular Automata*", Santa Fe Institute Studies in the Sciences of Complexity. Addison-Wesley
- [10] Burraston, D. (2006) *Cellular Automaton Data Tables*, <http://noyzelab.com/research/research.html>
- [11] Wuensche, A., (1997), "Attractor Basins of Discrete Networks", Cognitive Science Research Paper 461, Univ. of Sussex, D.Phil thesis
- [12] Wuensche, A. (2013) *DDLab*. <http://www.ddlab.org>

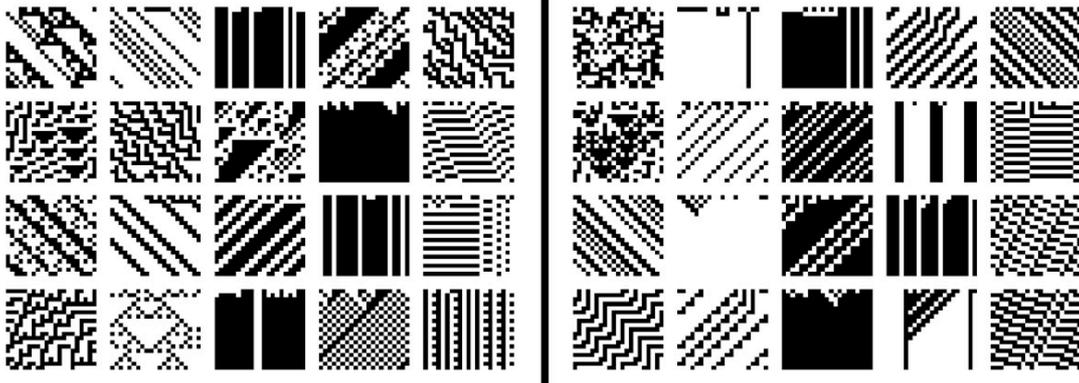
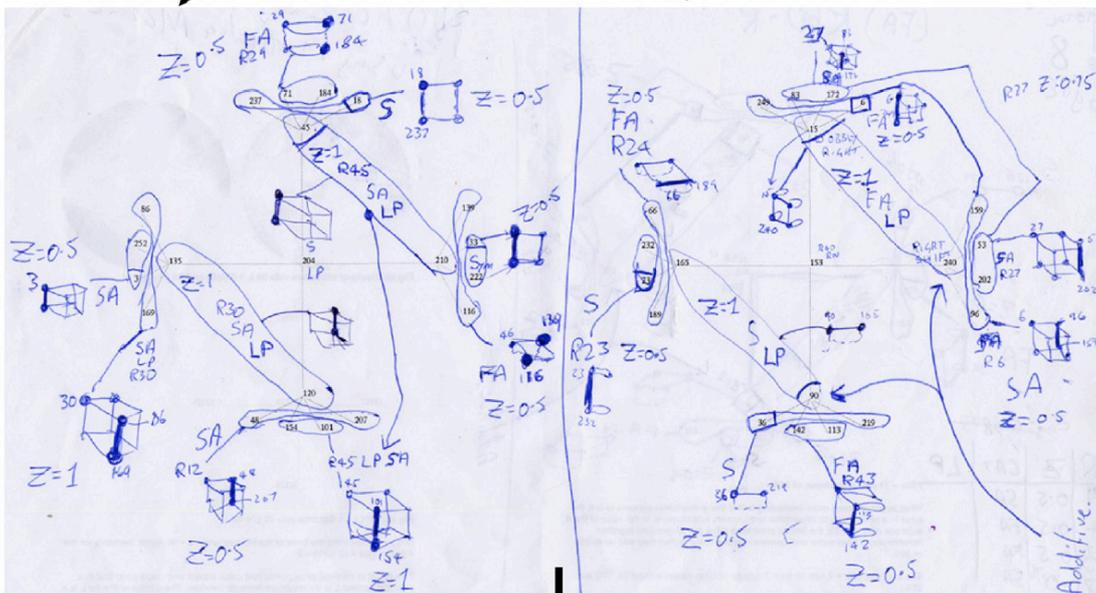
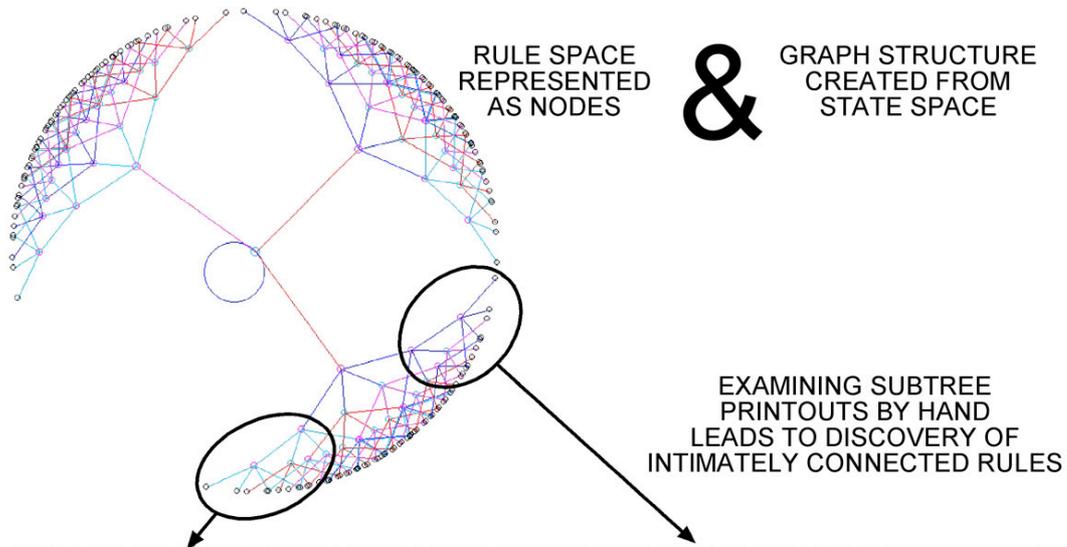


Fig 1. Using DDLab to create and annotate sub tree printouts of rule 90 with 8 cells resulted in the discovery that CA can self-organise rule space.