



## System Dynamics

**Abstract** This chapter introduces System Dynamics, a well-established method to developing full simulations of dynamic systems. We introduce the method and how it can be used to move from conceptual models to ‘stocks and flows’ models and simulations. We describe the steps involved in doing System Dynamics and consider common issues and tricks of the trade. We reflect on what the method is good and bad at and present a brief history of use and debates in the field. Finally, we give some advice on getting started yourself and point to some useful resources.

**Keywords** System Dynamics • Dynamic model • Stock and flow model

System Dynamics is probably one of the most widely known methods in this book, owing to its use in the Club of Rome commissioned report, *Limits to Growth* (1972). Though the report’s modelling and conclusions were heavily debated and criticised, there is no doubt that it explored the dynamics of economic and population growth within the constraints of the natural world in a compelling and engaging way—it showcased the power of System Dynamics, and modelling more generally, both as analytic and thinking tools.

This high-profile example reflects something more fundamental about the method; on a surface level, System Dynamics models are intuitive and often convincing. As a simulation method which produces results over

time (results which can be interpreted as forecasts, or even predictions—more on this later), it can create a black-box, magic-like quality; the lure of the simulated world convincing people of its truth. On a deeper level, however, discerning model observers and users can see the sensitivity of model results to an array of assumptions, and how the model can be used and abused by those wishing to use it to make their own arguments. Ultimately, though much energy was used on repetitive debates around Limits to Growth, and the model itself was criticised heavily, it likely did serve its purpose well—to raise awareness of an issue, and to facilitate and support more detailed thinking and strategic insight on it. Perhaps at a cost of System Dynamics itself, it also raised much discussion and thought on the different ways to model complex systems.

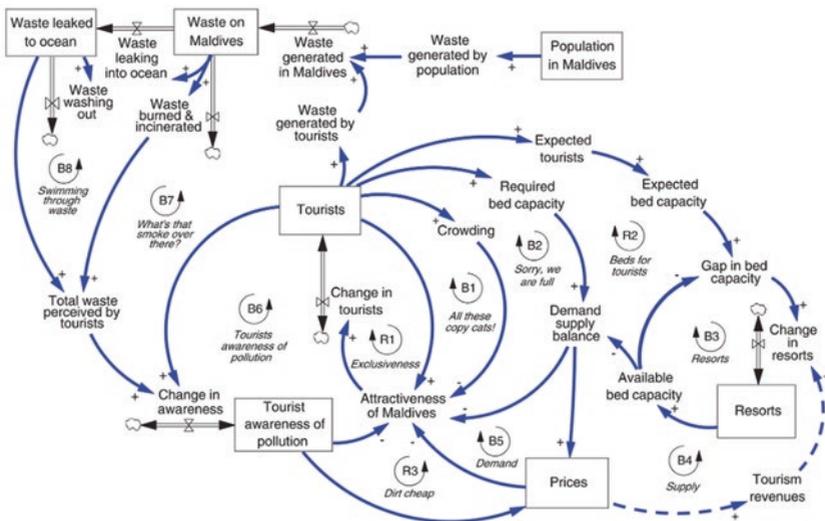
The rest of this chapter follows a now familiar structure. We first describe as clearly as we can, what System Dynamics is, before outlining how it is done. In these sections, we try to capture the nuances in differences in practice and how these affect what is produced. We also cover tricks of the trade and common issues you might come up against. Next, we try to summarise what System Dynamics is good at, and what it is not so good at, and consider how people tend to react to it. Finally, having introduced the method, we outline a brief history, so readers can get a strong sense of the points of debate and the trajectory of the method's use.

## WHAT IS SYSTEM DYNAMICS?

The short but jargon-filled definition of System Dynamics (many variants of which you will find in the literature) is 'a computer simulation method for analysis of complex dynamic systems'. If we unpick and decode this a little, we can say it (i) is typically implemented on a computer using software to carry out the calculations we put into the model, (ii) allows us to prod and poke a model to run different scenarios and look at different outputs for comparison, and (iii) is normally used to look at systems which are dynamic or moreover have some interesting dynamic properties.

More concretely, a System Dynamics model is made up of stocks and flows, and the factors which affect flows. Stocks are any entity which accumulates or depletes over time. A flow is the rate of change in a stock (usually represented by a differential equation). Stocks take numerical values, and flows are defined by equations which can be affected by any number of other factors in the model. These equations may be relatively simple, using standard operations like addition, subtraction, multiplication,

division, and sometimes exponents, or more complex involving functions and parameters which moderate how variables interact. Once a set of stocks, flows, and the factors which affect them are specified in equations, the simulation can be run by choosing a starting point (i.e. a set of stock values) and computing how stock values change through repeated time steps (i.e. through being repeatedly changed by their associated flow). Figure 8.1 shows an example of a stock and flow System Dynamics model which represents tourism and pollution in the Maldives (it is taken from Kapmeier & Gonçalves, 2018). As with most of the methods in this book, the diagram is in essence made up of boxes and arrows. Common to many System Dynamics diagrams, in Fig. 8.1 we can see the stocks as rectangles, the flows as straight hollow arrows with hourglass-type symbols (these are meant to represent valves) on them, and the factors that affect flows represented by floating text and the solid (and mostly curved) arrows. The small circular feedback arrows with italic annotations are labels for the feedback loops represented by the arrows around them. B stands for a balancing feedback loop, meaning that it will tend to self-stabilise, R for a reinforcing feedback loop, which will tend to carry on increasing or decreasing. Not so common, but used here, are the dotted lines to



**Fig. 8.1** A System Dynamics stock and flow diagram of pollution and tourism in the Maldives. (Source: Kapmeier and Gonçalves (2018))

represent influences which were not included in calibration processes due to missing data (more on calibration later).

To describe some of the equations in the model, let's take the example of the number of tourists. This is represented by the stock 'Tourists', which changes based on the flow 'Change in tourists', which is, in turn, determined by a constant historical tourist growth rate (we cannot see this in the diagram, but it is specified in text which describes the model) modified by the factor 'Attractiveness of the Maldives' (which we can see here). Essentially, the flow is a differential equation specifying the rate of change of tourists over time, as the historical rate of change is impacted by the changing attractiveness of the destination.

$$\frac{dT}{dt} = T * g_h * (1 + A)$$

where  $T$  is number of Tourists,  $g_h$  is Historical Tourist Growth Rate and  $A$  is Attractiveness of the Maldives.

'Attractiveness of the Maldives' is calculated based on five inputs: tourist numbers, crowding, demand/supply balance of beds, prices, and tourist awareness of pollution. The ways these are specified in an equation can vary from something simple to something more nuanced. In Kapmeier and Gonçalves (2018), they are calculated with the following equations:

$$\text{Attractiveness} = E_R + E_{DS} + E_p + E_p$$

where  $E_R$  = Effect of Resort Exclusivity on Resort Attractiveness;  $E_{DS}$  = Effect of Demand/Supply Balance on Resort Attractiveness;  $E_p$  = Effect of Pollution on Resort Attractiveness;  $E_p$  = Effect of Price on Resort Attractiveness

$$E_R = f_1(\text{Tourists} / \text{Tourists reference value})$$

$$E_{DS} = f_2(\text{Required Bed Capacity} / \text{Annual Bed Capacity})$$

$$E_p = f_3(\text{Pollution} / \text{Pollution reference value})$$

$$E_p = f_4(\text{Price} / \text{Price reference value})$$

where the  $f$  functions are decreasing s-shaped logistic functions, and reference values estimated from past data are used to normalise the input variables (i.e. dividing the current number of tourists by a reference value

normalises it relative to that reference). The logistic functions bound the effects on attractiveness of any variable between specific upper and lower limits, ensure that increasing values of numbers of tourists, demand/supply balance, pollution, and price decrease resort attractiveness and allow the system to be set up so that a high value of any one of these factors dominates over any positive effects from others. Similar sets of equations are described for each of the stocks and flows in the map and the influences on them. These equations together allow the dynamical behaviour of the system to be simulated.

It is often assumed that System Dynamics models are deterministic using the same exact values each time and thus producing the same result every time; however, this is not true. Many models, and indeed very early examples, included stochastic elements. What types of things you can represent in stocks and flows is in theory broad. We do not have to have data on things in the model, and they don't even have to be quantifiable. Indeed, the founding figures of the method were clear that we did not need to revert to using highly quantitative factors with data to back them up. As long as it makes some sense for stocks to be talked about going 'up and down' and we can put some numbers to them and factors in equations without it being completely meaningless or inappropriate, we can make a model work.

However, in practice, people often favour more quantifiable elements and avoid 'soft' (unquantified) factors. This is perfectly understandable; the existing literature and data make it much easier to quickly implement a biophysical factor such as 'rainfall', for example, compared to a social factor such as 'Attractiveness of the Maldives' (as above). The natural and physical sciences tend to have more data and quantitative models of almost all the factors we might ever wish to use from these domains, whereas the social sciences do not. This data and models make it easy to implement these factors (i.e. we know their values, and we know what they are affected by). We must put in extra effort and take extra care to include important social and 'soft' factors in our System Dynamics models. Specifying mathematically how they interact may take a great deal of careful thought. All complex adaptive systems will have these components, and they will often play important roles in dynamics, so we cannot ignore them simply because they are more difficult to specify.

Once a System Dynamics model is built, it is 'run', this produces outputs, such as those seen in Fig. 8.2. Here we see values for the stock 'Tourists', through time, under multiple different scenarios (labelled

capacity, access, and price, there is also a baseline scenario and some real data plotted). A scenario represents a change in the model, either in the equations of flows or in the structure of the model (i.e. with a connection added or removed), both of which are intended to represent some intervention or difference in the system. Here the scenarios represent different policy options. Quite different sorts of system futures are visible under different scenarios. For example, increasing tourist access or capacity ultimately leads to a crash in tourist numbers as the resulting pollution from high tourist numbers destroys the destination's attractiveness. A policy of lowering prices does not have such large impact on tourist numbers and hence does not lead to a boom-and-bust dynamic.

There is ongoing debate in the System Dynamics community, and other simulation communities, as to whether we should ever (try to) interpret model outputs as predictions of future states. Many scholars contend that as the models are simplifications of reality, and the reality we are modelling is a complex adaptive system with many components and interactions, models will never be able to provide reliable predictions of future states, so we should not attempt to use them in this way. Rather, they suggest, we should consider model outputs as hypothetical futures, or qualitative forecasts of the types of patterns and dynamics we might see. However, others take an opposing view, seeing prediction as one of the main purposes of models, and while acknowledging the difficulty of prediction in complex systems, still wish to attempt to predict using models like System Dynamics,

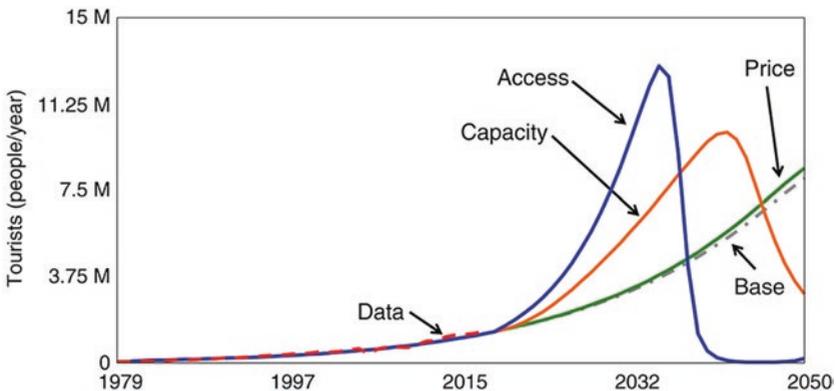


Fig. 8.2 Number of tourists under different policy scenarios. (Source: Kapmeier and Gonçalves (2018))

and (partially) assess the quality of models against their ability to predict. Without some element of prediction, they see models as missing a key element of what defines a ‘good’ model. We tend to sit on the fence on these debates, taking a pluralist view and seeing the merit in both approaches. As we have often said in this book, purpose drives the model. If our purpose is to predict, and we have designed with this in mind, it is not beyond the reach of methods like System Dynamics, but it is certainly very difficult, and success may well be fleeting.

As with many quantitative modelling methods, there are a range of rather technical activities that will be part of any System Dynamics process, which include parameterisation, calibration, validation, and sensitivity analysis. In practice, these elements often merge into one another, and the exact nature of how they are done, in what order and combination, and their importance in the modelling process, will depend on the purpose of each modelling project. We do not wish to go into the gory details of how they are done, which is beyond the scope of this chapter, but briefly, here are some simple (yet still debatable) definitions of them: parameterisation is the process of picking values for parameters (e.g. stocks, factors, equations, modifying flows) in the model based on data or plausible values (i.e. ‘plausible’ is a loose idea, but here we mean values which an expert on the system would deem sensible); calibration is the process of picking values for parameters such that the model produces outputs that reproduce some target (e.g. changing values until the model produces a pattern seen in real data); validation is the process of assessing how well the model reproduces target outputs (and often involves elements of calibration); and sensitivity analysis is where different parameters, or structural components of the model (e.g. which connections and relationships are represented), are changed systematically to assess how sensitive model outputs are to them. Each of these activities can be done in informal qualitative ways, or in much more technically demanding quantitative ways.

There are some important variations in practice and terminology in System Dynamics. In the practice of System Dynamics modelling, these mostly revolve around how we arrive at a fully specified and quantified stock and flow diagram. Some system dynamicists dive straight into a stock and flow diagram, whereas others first produce Causal Loop Diagrams (see Chap. 4) or other similar types of qualitative or semi-quantitative systems maps. These differences in where to start will likely have impacts on the models produced. Causal Loop Diagrams are a point of variation in terminology too. We have observed Causal Loop Diagrams referred to

widely as ‘system maps’. The way they are built appears to be consistent, but the different terminology can be confusing and unhelpful when the term ‘system maps’ also describes a variety of approaches, as we outline in this book. We have also seen stock and flow diagrams referred to as Causal Loop Diagrams and vice versa, though their intended use is normally clear.

Before we move on to how to do System Dynamics, we would like to emphasise one element in the method that is different to some of the others in this book but is not immediately obvious. Despite having ‘system’ in the name, System Dynamics models tend to be focused on specific dynamical problems rather than whole systems. In practice, this means that models rarely aim to represent a whole system, but rather focus on carefully constrained sub-sets of a system with the relevant dynamics. This nuance in focus is often not understood, with many researchers and practitioners envisaging System Dynamics as taking a whole-systems view, where in fact it is the ‘dynamics’ that the method emphasises. Indeed, part of the ‘art’ of System Dynamics modelling is finding and defining these knotty dynamical questions and problems which the method can contribute to significantly.

## HOW DO YOU DO SYSTEM DYNAMICS?

The process of conducting a System Dynamics modelling project can be broken up into six steps. These make sense conceptually, but are never actually done neatly in sequential order, rather much iteration and jumping backward and forward is the norm. They are:

1. **Problem definition:** to start, we need to pick a system and a dynamical problem or question within it. This stage is arguably the most difficult, and the one for which your approach will change and improve most as you gain experience; new system dynamicists often want to model a whole system, or struggle to find a dynamic situation where the modelling will work best. Problem definition is often done in partnership with stakeholders or users of the model and research. Taking their views on board and implementing them in the project and model is difficult, as with all methods, but particularly so with System Dynamics because it is one of the most formalised, and thus potentially rigid, of the methods in this book.
2. **Model conceptualisation:** this is the stage at which we begin to map out the dynamic problem in something that resembles a ‘boxes

and connections' model. Most commonly this is done with Causal Loop Diagrams (see Chap. 4 for an introduction and description of how to use this method, we don't reproduce it here in the interests of avoiding repetition) but may also be done with other qualitative systems mapping approaches akin to Participatory Systems Mapping (see Chap. 5) or Fuzzy Cognitive Mapping (Chap. 6). The point is to use something which is easy to build but which also provides the building blocks for the quantified model of stocks and flows.

3. **Model formulation:** in this stage we take our conceptual model and turn it into a fully specified and quantified stock and flow model. The process for doing this is one of judgement and modelling choices (of which there will be many), rather than a simple conversion or standardised process. You will need to decide what elements should be stocks, which should be flows, and which should be factors that affect flows. Finally, you will need to specify the equations which determine the flows. These are often simple but can involve quite advanced mathematics if you so wish.
4. **Validation and simulation:** once the model is fully built, it is time to 'run' it (i.e. crunching the numbers, letting the stock values and flow equations play out). Normally, you do this multiple if not many times, systematically varying the model design or parameters. This is done either as part of calibration and validation, or to represent the different scenarios you might wish to explore, or as both, in many iterations. We suggest specifying a set of experiments to help you think thoroughly about different model runs you wish to do, and help you communicate this. You may require longitudinal data on the system or problem you are modelling, against which to compare your model results. This stage often takes much longer than people imagine, and uncovers issues in the model build, conceptualisation, or even in the problem definition, that require returning to those stage to fix—this is healthy iteration and should not be avoided. Your model will produce reams and reams of simulated data, normally in CSV (comma separated values) file format.
5. **Analysis:** once you have your model outputs, and any real data you are going to use, there will be a significant amount of time spent on analysing, understanding, and presenting those results. You should follow good practice in terms of planning and recording your analysis so it can be reproduced by you or others at a later stage. Equally, you are likely to want to think carefully about how you visualise

outputs and should look at similar examples you think work well, or follow more general guidance on data visualisation.

6. **Interpretation:** finally, once we have results we can share (and importantly, our descriptions and explanations of them), it is time to interpret and use these results with stakeholders and end users. Again, this is a key and often underestimated stage. The most impactful and successful projects will give plenty of time for results to ‘stew’ and be interpreted and allow space for further iteration from there.

The exact detail of each of these steps and how you iterate within and between them will depend on many factors, most importantly your own preferences and the purpose of the project. We cannot emphasise this enough: do not be worried about jumping back and forward, the best modellers do, and are creative as they go.

As with most of the methods in this book, System Dynamics modelling can be done in a participatory setting, and this is common. Sometimes it is referred to as ‘group model building’, ‘mediated modelling’, or ‘Participatory System Dynamics’. A key decision is what stages of the modelling to do with stakeholders; do we just do the problem definition and model conceptualisation, then return with results, or do we also include them in the model formalisation stage? Both will be valid depending on your purpose. We would encourage you to attempt to include stakeholders at all stages, and not assume they do not want to get into technical details. The last thing we want in a participatory modelling project is for stakeholders to think there are ‘highly technical and skilled’ elements which they should not play a role in and thus passively wait for your return ‘with the answers’. It may go against our instincts, but ideally, we want stakeholders to critique and attack our models, so we can uncover bad assumptions, misunderstandings, and the gaps that hamper consensus being reached. The trick is to have just enough legitimacy and respect with stakeholders so that they will take part and engage, but not so much that they won’t be sceptical and criticise you.

There are many software options available for a budding system dynamicist. Point-and-click options are common, and you will not need to write the computer code for a model if you do not want to. Some of the most popular and powerful include (in alphabetical order):

- **NetLogo**: this free and open-source agent-based modelling software also has an easy-to-use System Dynamics functionality.
- **iThink and Stella**: these are two product names for a range of commercial System Dynamics software produced by isee systems.
- **Powersim Studio**: commercial software, with some free versions available.
- **Vensim**: commercial software, with some free versions available.

Though it is no longer in use, the DYNAMO simulation language is worth a special mention. This was the language used for many early System Dynamics models, including the Limits to Growth model, and is still spoken about with affection. Some efforts exist to bring it back to (some form of) life, for more, see <https://github.com/bfix/dynamo>.

### COMMON ISSUES AND ‘TRICKS OF THE TRADE’

We touched on one of the more fundamental but common issues above—struggling to find a dynamical problem or question which is well-suited to a System Dynamics model. We often begin a project with a broad topic or interest, and perhaps a method in mind, and need to find a way in which the latter can be of use in the former. This is a difficult art. Some rules of thumb for defining the ‘target’ of the model (i.e. the thing we want to model) include:

- Avoid trying to model a whole system.
- Look for dynamical sub-parts of a system and look for issues or problems which persist despite the presence of dynamics and interventions (i.e. a problem which persists because there is no intervention or any dynamics affecting it is not really a dynamical problem).
- Ask for advice from more experienced modellers and more generally share and air your ideas regularly.
- Think about what the outputs might look like—can you sketch the graphs over time your model might produce?
- Dive into some modelling knowing you are likely to iterate back to your ideas to update them as you bump into issues building a model.

It is relatively easy to start building System Dynamics models, but much harder to produce ‘good’ models and develop genuine insights. This can create a common sense of frustration for intelligent but inexperienced

modellers. While the technical craft can be picked up quickly, you will likely bump into many conceptual issues and questions which are more difficult to ‘solve’, and you will also start to see a range of issues and weaknesses in your model and the purpose you are putting it to. While as modellers we are often told not to ‘fall in love’ with our model, it can be just as common and problematic for new modellers to become disillusioned at the difficulty of good modelling and lose interest. This is normal and part of the learning process. This can happen with any of the methods in this book, but it is particularly relevant here because System Dynamics is a simulation method with many components to be specified, calibrated, a model to be validated and scenarios to be designed and run—in short, there are simply more points at which to run aground.

One of the keys to avoiding this issue, and many others, is to iterate quickly and aggressively. You will see calls for iteration everywhere in research and modelling, and especially for work on complex adaptive systems, where iteration is a key heuristic in researching and engaging with systems. But how do you actually do it? What does it look like? Most simply, it means jumping backwards and forward between the six steps we describe above. There is a difficult balance to doing this though, it can be done too slow and too fast. The trick is in having the confidence to move between steps openly and with purpose, but to do so in a systematic way rather than in a panic.

### WHAT IS SYSTEM DYNAMICS GOOD AND BAD AT?

System Dynamics’ real strength is in helping us develop high-level strategic insights on dynamical questions. It can help us find counter-intuitive dynamics, to rigorously play out the implications of an intervention or theory, or to play a facilitating role in doing this with stakeholders. It also helps us think about dynamics rather than individual events or static models. As well as producing outputs showing how variables change dynamically, many systems dynamics software packages allow you to see how dynamics emerge from interacting feedback loops as they change in strength and importance as a model unfolds. This breaking down of dynamics into component parts can allow more intuitive understanding of how dynamics arise for stakeholders and modellers alike. Thinking in a dynamical way is a very different way of thinking for many people, which can be hard to develop; System Dynamics will help you do it. As a simulation method, it is also well-suited to helping us test out assumptions and

the potential impacts of interventions; to exploring those ‘what if?’ questions.

Despite being a quantitative method, it is not useful for operational or more local, micro, questions, or more generally in detailed analysis of systems. It normally represents systems at an aggregate level and focuses on dynamical subsystems. It requires a relatively large amount of time to conduct compared to other methods in this book. This can be a serious barrier to its use in applied and participatory contexts. It can also be difficult to include social and ‘soft’ factors in a model, where we don’t often have the help of previous modelling and quantitation of these ideas to inform our implementation of them.

### A BRIEF HISTORY OF SYSTEM DYNAMICS

The method appeared in the 1950s, developed by Jay Forrester, who used it to explore the success and failure of businesses and industrial dynamics. Legend has it, Forrester used ‘hand simulations’ (we can only assume these were calculations done on paper, rather than physical representations of the stocks and flows!) to explore employment dynamics with managers at General Electric. Forrester and his team quickly moved on to modelling on computers, and his approach was applied to multiple business and management questions. By the late 1960s and into the 1970s, the method started to spread beyond these roots and was famously used in the Limits to Growth work described above. Since then, it has been applied in many domains and enjoys a large and active community of practitioners and researchers.

Because of the method’s name, and partly because of its approach, people often assume there are strong connections with systems science and thinking; however, the method did not emerge out of these fields, so the links can sometimes be weaker than we might expect or hope. Nonetheless, the underpinning interest in complex adaptive systems is present in both. Prominent figures in the field have suggested progress has been held back by the lack of connection to other modelling and research domains, and this has led to a lack of development in the method itself (e.g. Sterman, 2018).

There have been, and continue to be, many debates in the community. They have revolved around questions such as: whether we should aim to make point predictions; whether we have to actually build a simulation, rather than just a static model, to be ‘doing’ System Dynamics; and what

types of validation are appropriate and how quantitative they should be. There are also different schools of thought on how the method should be used, either as a participatory tool, or as a purely analytical tool. All of these sorts of debates are common in simulation methods, though they do appear to be had rather more forcefully in this community as opposed to others. We believe there should be room for all the different ways of doing System Dynamics to exist under the one umbrella. Modelling should be driven by purpose, and as long as there are different purposes behind System Dynamics projects, there will be justifications for doing it in different ways; with or without a simulation, with stakeholders or as a highly academic endeavour; with modern quantitative validation methods, or with more qualitative and heuristic validation.

### GETTING STARTED WITH SYSTEM DYNAMICS

Because System Dynamics is somewhat more formal than some of the other methods in this book, we are little more cautious than our usual ‘dive in and see what happens’ approach to getting started. The benefits of doing some more reading, and maybe even attending a course, are likely to be high for this method.

Perhaps, the most obvious place to start is the System Dynamics Society. The society was setup in 1983 with Forrester as president and has continues to be the main voice and organising force in the field. Its journal—System Dynamics Review—is one of the main places to find current work using the method, and its annual conference is worth considering too. Its website is easy to get lost in, has a plethora of resources, and can point you to up-to-date information on events and training, including an online course catalogue, readings, and videos.

We would recommend looking at some of the following further reading and resources:

- Loopy (<https://ncase.me/loopy/>) is an extremely easy-to-use and fun browser-based quasi-System Dynamics tool. Though its functionality is limited to the very basics, it is great tool to get a quick sense of how modelling dynamical systems in causal networks feels. It can be used to play around with, or to build and share simple models.
- The special issue in 2018 of the System Dynamics Review ‘Celebrating the 60th anniversary of the System Dynamics field’ is a good place to

start on more academic and detailed pieces. John Sterman's introduction is an excellent, if very strong, description of some of the history of the field and a call to arms to improve practice and link with other domains.

- For those who like to return to the classics of a field, *Industrial Dynamics* by Jay Forrester (1968), and *Business Dynamics* by John Sterman (Sterman, 2000) are the obvious choices, and both still considered relevant to budding and experienced system dynamicists.

Despite the need to take a little more care with System Dynamics than some of the other methods in this book, we hope you are not discouraged. You can still easily have a play with some of the software above or online models such as isee's COVID-19 simulator (<https://exchange.iseesystems.com/public/isee/covid-19-simulator>). Dip your toe in and see how it takes you. The joy of producing some dynamic output can really capture your imagination. It is also worth emphasising, it is fine to be self-taught in more formal methods like System Dynamics. Most researchers and practitioners are. Don't be put off by the fact you are not going to enrol on a masters or PhD programme in System Dynamics. If nothing else, giving System Dynamics a go will help you to understand what it means to 'think dynamically'. In our experience, this is a rare, difficult, and valuable skill.

## REFERENCES

- Forrester, J. W. (1968). *Industrial dynamics*. Productivity Press.
- Kapmeier, F., & Gonçalves, P. (2018). Wasted paradise? Policies for Small Island States to manage tourism-driven growth while controlling waste generation: The case of the Maldives. *System Dynamics Review*, 34(1–2), 172–221. <https://doi.org/10.1002/sdr.1607>
- Meadows, D. H., Meadows, D. L., Randers, J., & Behrens, W. W., III. (1972). *Limits to growth. A report for the Club of Rome's project on the predicament of mankind*. Universe Books.
- Sterman, J. D. (2000). *Business dynamics: Systems thinking and modeling for a complex world*. McGraw Hill.
- Sterman, J. D. (2018). System Dynamics at sixty: The path forward. *System Dynamics Review*, 34(1–2), 5–47. <https://doi.org/10.1002/sdr.1601>

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

