# EMBER: Emergence and Macroscopic Behavioural ExtRaction User Guide

Spencer A. Thomas spencer.angus.thomas@gmail.com

May 13, 2018

# Contents

1	An	introduction to EMBER	<b>5</b>
	1.1	What is EMBER?	5
	1.2	Download and instillation procedure	6
	1.3	Running the code through a terminal	6
	1.4	Parameter File	6
	1.5	A word of warning	6
	1.6	Configuration phase	6
	1 7	Extending EMBER	7
	1.1	Contact information	7
	1.0	Structure of this guide	7
	1.9		1
2	Sto	chastic models	9
	21	Stochastic double well	9
	2.1	2.1.1 The Model: Details	g
		2.1.1 The Model. Details	10
		2.1.2 Automa of Equation free Analysis	10
	0.0	2.1.5 Outcome of Equation-free Analysis	10
	2.2	D.2.1       The Model: Details	10
		2.2.1 The Model: Details	10
		2.2.2 Requirements of the User	10
		2.2.3 Outcome of Equation-free Analysis	ιL
2	Ago	ant based models, simple models	2
0	Age 2 1	Spread of fire in a forest	.0 19
	3.1	2.1.1 The Medel Ar Occurring	10 10
		<b>0.1.1</b> The Model: An Overview       1 <b>0.1.2</b> The Model: An Overview       1	10
		3.1.2 The Model: Details	13
		3.1.3 Requirements of the User	14
		3.1.4 Outcome of Equation-free Analysis	4
	3.2	Ising model: a generalised transition model	15
		3.2.1 The Model: An Overview	15
		3.2.2 The Model: Details	15
		3.2.3 Requirements of the User	16
		3.2.4 Outcome of Equation-free Analysis	18
4	Age	ent-based models: complex models	11
	4.1	A model of altruism	21
		4.1.1 The Model: An Overview	21
		4.1.2 The Model: Details $\ldots$	21
		4.1.3 Requirements of the User	22
		4.1.4 Outcome of Equation-free Analysis	23
_			_
5	Add	ditional Information	17
	5.1	Publications	27
	5.2	Future work	27
	5.3	External algorithmic developments	27
	5.4	Further Reading	28
C	1		
G	ossa	гу	:9

#### Acronyms

### Chapter 1

## An introduction to EMBER

#### 1.1 What is EMBER?

Emergent and Macroscopic Behavioural ExtRaction (EMBER) is a generalised analysis tool for numerical and dynamical systems analysis of non-linear stochastic models or simulations. By simulation, here we mean any 'black box' that takes an input and produces some form of output after some unknown processes. This includes, but is not limited to; agent-based models (ABMs), computational models, equations of state, stochastic models or physical experiments.

EMBER uses mathematics techniques such as numerical continuation, bifurcation analysis and equationfree methods to extract detailed information regarding the systems behaviour directly from the simulator. EMBER does not require a analytical description of the system (i.e. a closed form solution) and can be applied to an input/output 'black box'. EMBER can also be used to analyse the macro level behaviour of models where there only have analytical descriptions of microscopic behaviour (e.g. stochastic equations). Those interested in the technical details of how EMBER are referred to [23] for the specifics of this program, and should consult the list of publications in §5.1 for a broader overview of these techniques.

EMBER uses a technique known as equation-free techniques that enables it to perform tasks such as,

- determine macroscopic behaviour
- parameter dependence
- bifurcation analysis
- sensitivity analysis
- path-following
- tipping point analysis
- identify drivers parameters
- locate critical pointss
- determine regimess and their boundaries
- statistical analysis such as heteroscedasticity
- distributions of variables at each parameter step
- build 'road maps' of the behaviour of the system
- and many others

Several of the examples in this user guide are for ABMs, though EMBER is not limited to these and can be extend to other models and simulators. We include some of these other models and citations to the use of equation-free methods in other systems. The code is java based and is compiled and run through the command line. The necessary commands are given in the section . A user interface will be developed in the future.

#### **1.2** Download and instillation procedure

The current version of the code is available for down load from https://www.dropbox.com/sh/r4w33ao3larqbev/ AAA\_fP\_U4QCZ-5i\_4Ykj1xM\_a?dl=0 which contains all the required directories and sub folders. In order to run the code, Java (https://java.com/en/download/) and Netlogo (https://ccl.northwestern.edu/ netlogo/5.0.5/) must be installed. Currently EMBER has only be tested on netlogo version 5.0.5, newer versions may cause errors due to the API between Java and NetLogo. A runtime error is displayed if the version of NetLogo is not compatible.

#### 1.3 Running the code through a terminal

Mac and Linux users can run the code through a terminal window by navigating to the parent folder and typing

./compile.sh to compile to the code, then

./run.sh

to run the analysis. These are simple bash scripts for convince and may be usable on windows but this has not been tested. Note the *jar* file can be executed on windows directly. If the you receive an error when running the above scripts saying permission denied you will have to enable by typing;

 $chmod \ u+rwx \ compile.sh$   $chmod \ go+rx \ compile.sh$   $chmod \ u+rwx \ run.sh$   $chmod \ go+rx \ run.sh$ then re-run the ./compile.sh and ./run.sh scripts.

#### **1.4** Parameter File

The file *ContinuationParameters.java* lists all the required parameters and comments to explain what they are as well as any default values. Parameters include those required by the model to be analysed, and those required by EMBER. The latter include some tunable parameters that can be manually determined by the user or using EMBERs automated testing phase which will provide some predictions for these values for unfamiliar users or models. Please note that this requires some iterative analysis and running of the model which can take a long time if the model has a large computation time. The computation time can also increase for a large number of realisations of the model, however, EMBER uses a convergence constraint correct repeat algorithm introduced here [23] which enables robust analysis even at a small number of realisations. This is vital for systems with high levels of noise or large run times,

#### 1.5 A word of warning

It is important that the *Lift Operator* and *Restrict Operator* are inverse operations. That is if performing the *Lift Operator* followed by the *Restrict Operator* without running the simulation must yield the same state, otherwise the analysis may not work correctly. For analysing stochastic models this is straightforward, and may only become problematic for 'black box' models.

#### **1.6** Configuration phase

A requirement of EMBER is to define the parameters; Realisations, Number of Steps, Time Horizon, Continuation Step and the direction of this step Continuation Step Sign. The Number of Steps is simply how many points you want to calculate, i.e. if you want to go from 0 to 1 in steps of 0.01 then this would be 100. In this example 0.01 would be the Continuation Step and the direction would be +. If you wanted to go from 1 to 0 then Continuation Step would still be 0.01 and Continuation Step Sign would be -. The Time Horizon is the length of the time simulation and Realisations is the number of independent simulations in the ensemble used by EMBER. The Number of Steps and Continuation Step Sign are in complete control of the user. The other parameters can be either be select by trial-and-error or intuition, or using our configuration phase within EMBER. In some cases it may be appropriates to use insight to define these parameters. For instance if we are looking at the percentage of a quantity we have a defined range of 0 to 100 and may chose our Continuation Step based on how precise we wish to obtain our results; a step of 1 may be sufficient for our needs, or a smaller step systems with a strong dependence. Having a smaller Continuation Step requires a higher Number of Steps for a given range thus will increase the computation time so the user may wish to 'play' with this value. Like wise the number *Realisations* may be easy to define with some observations of the model, highly variable outcomes will require more *Realisations* and models that experience lower levels of stochasticity can have fewer. The *Time Horizon* is harder to define. Due to this and the ambiguity with selection of the other parameters, and the problem specific nature of their values, we have defined a systematic method to obtain them based on the dynamics of the model. In this configuration phase, EMBER simulates the model under a parameter setting provided by the user to obtain the values for the *Time Horizon, Realisations* and *Continuation Step*. Users can take these values verbatim or as a guide for reasonable values. That is, the values provided are sufficient for EMBER to successfully analysis the model, however, the user may require higher precision results though more *Realisations* or wish better resolution of the macroscopic dependence curves through *Continuation Step*. The details of this part of EMBER are beyond the scope of this user guide, however readers are interested should refer to the published material in  $\S 5.1$  which contains details of this, and other parts of EMBER.

#### 1.7 Extending EMBER

This feature of Emergent and Macroscopic Behavioural ExtRaction (EMBER) is for advanced users who are comfortable with computer programming and Java. Extending EMBER to other programs and simulators is straight forward due to the modular design of the underlying implementation of the tool. Extending EMBER to simulators, including ABMs in languages other than NetLogo, simply requires the editing of one program 'module', *MacroOperator.java*, in the source code. This is the only part of EMBER that interfaces with the model or system of interest, and can therefore be edited in isolation of the rest of the code due to the object oriented design. The underlying implementation of EMBER is in Java and the application program interface (API) is defined in *MacroOperator.java* only. Therefore as long as an API can be created between EMBER and the system of interest then defining this API in *MacroOperator.java* will enable EMBER to analyse the system in the same was as the examples in this guide.

#### **1.8** Contact information

In addition to this guide information on EMBER is available at; http://steerplex.org.uk/tools/tools-advanced/equation-fee-numerical-continuation-tool Contact information is given below.

Topic	Person	Email
General enquiries	Spencer Thomas	spencer.angus.thomas@gmail.com
	Anne Skeldon	a.skeldon@surrey.ac.uk
	David Lloyd	d.j.lloyd@surrey.ac.uk
SteerPlex	Pete Barbrook-Johnson	p.barbrook-johnson@surrey.ac.uk
LimnoSES model	Romina Martin	romina.martin@su.se

#### 1.9 Structure of this guide

The guide is by no means exhaustive and examples are added as and when they can be. The structure of the guide is broken up into chapters to help users identify systems that closely resemble theirs. In Chapter 3 we include simple examples as an introduction to the how EMBER can be used to analyse models and what kinds of insight can be generated. Many of the models in this chapter are open source. In Chapter 4 we look at more complex models, many of which are designed to model real world systems. The application of EMBER here is much more involved and this chapter is intended to be a guide to practical applications of the tool. In Chapter 2 we go beyond agent-based models (ABMs) and demonstrate how EMBER can be applied to other systems to extract insight in the same way. Finally in Chapter 5 we include some additional information such as publications relating to EMBER, future work and further reading on some of the topics and examples in this guide.

### Chapter 2

### Stochastic models

#### 2.1 Stochastic double well



Figure 2.1: Double well potential for differing values of barrier height ( $\mu$ ) as indicated on the *y*-label and parameter dependence curve for the system. Stable (unstable) states are indicated by black (white) circles and solid (dashed) lines.

#### 2.1.1 The Model: Details

The stochastic double well is modelled with a potential;

$$\frac{dV(x)}{dx} = -x(x^2 - \mu) - \nu , \qquad (2.1)$$

where  $\mu$  is the barrier hight and  $\nu$  is the asymmetric term in the well. The state of a particle in the well is given by;

$$\dot{x} = \frac{dV(x)}{dx} + \eta \dot{W} , \qquad (2.2)$$

where  $\eta$  is the noise level and W is Gaussian noise between 0 and 1. In this analysis we set  $\eta = \sqrt{dt}$ , where dt is the step size in the ordinary differential equation (ODE) solver.

#### 2.1.2 Requirements of the User

As we have the microscopic description of this system, we can supply the values for the model parameters, here barrier-height  $\mu$ , asymmetry term  $\nu$  and the level of noise  $\eta$ , and use EMBER to obtain the macroscopic behaviour of this system, which is not obtainable with classical mathematical techniques. This demonstrates the flexibility of the EMBER tool as this model can be implemented in a number of different way, such as in a Java description or implemented as a Netlogo ABM, and obtain the same results from the analysis. This gives the investigator the option to analyse a system coded in their preferred language or to compare other implementations of similar models without having to implement them. An example of the how a used we set up the EMBER analysis of this model with our user interface is illustrated in Fig. ??. The model has only one variable, x the position of the 'particle' in the well. Taking the model parameters as  $\mu = 3$ ,  $\nu = 0.3$  and  $\eta = 1$  we take an initial guess at the value of x = 1.75 and perform our analysis by investigating the dependence of the macroscopic behaviour of the system, i.e. x, with the barrier hight  $\mu$ . As there is only one variable in the system, our *Lift Operator* and *Restrict Operator* are simply defining and calculating x respectively. The model can be analysed using the following settings.

#### 2.1.3 Outcome of Equation-free Analysis

We focus on the upper left curve in Fig. 2.1 due to the presences of a tipping point. Using the system exploration phase to determine the essential continuation parameters we obtain N = 100,  $\tau = 1.0$  and  $\delta s = 0.05$ . With these parameters we use our algorithm to perform Equation-free (EF) analysis of this model following the path around the tipping point indicated in Fig. 2.3. Note the successful analysis past the critical point, which is a common problem in this type of analysis. EMBER analysis uses algorithms to specifically enhance its robustness to enable the investigation of highly stochastic models without requiring a large ensemble of simulations which may increase computational times significantly. We note that previous analysis of this system in [2] used an ensemble of  $10^4$  simulations, whereas here we use on N = 100 and obtain the same results.

Analysis of this stochastic system has successful revealed the system's parameter dependence on the barrier height  $\mu$ . Moreover, it has correctly identified the tipping point observed in the deterministic case despite the models stochastic nature. Additionally, EMBER has correctly identified the difference between the upper and lower paths, coloured red and blue in Fig. 2.3 respectively. This demonstrates that EMBER can perform such analysis in the presence of models with high variability between simulation runs, and in cases where a deterministic system is unknown or does not exist, EMBER can extract the macroscopic behaviour.

#### 2.2 Hysteresis

#### 2.2.1 The Model: Details

A system exhibiting hysteresis can be modelled by

$$\frac{dy}{dx} = x(1-x)^2 + \lambda + \eta \dot{W}$$
(2.3)

where  $\lambda$  is a bifurcation parameter,  $\eta$  is the noise level and  $\dot{W}$  is Gaussian noise between 0 and 1. The inclusion of the noise term prevents the use of traditional numerical continuation of this system.

#### 2.2.2 Requirements of the User

As before we have the microscopic description of this system and can supply the values for the model parameters,



Figure 2.2: Continuation of a stochastic double well potential is compared to the deterministic case (right) with where stable (unstable) branches are illustrated by red (blue) points. Error bars are  $\pm$  two standard errors,  $\tau = 1.0$ , N = 100 and  $\delta s = 0.05$ .

```
String[] systemParameters = {"lambda ", "eta "};
double[] param = {-3.0, 1.0};
double[] Initial = {2.2};
boolean isSystemInitialised = true;
```

#### 2.2.3 Outcome of Equation-free Analysis

Analysis of this stochastic system has successful revealed the system's parameter dependence on  $\lambda$ . Moreover, it has correctly identified the tipping points observed in the deterministic case despite the models stochastic nature.



Figure 2.3: Continuation of a stochastic hysteresis model. Error bars are  $\pm$  two standard errors,  $\tau = 0.01$ , N = 500 and  $\delta s = 0.01$ .

### Chapter 3

### Agent-based models: simple models

#### 3.1 Spread of fire in a forest

#### 3.1.1 The Model: An Overview

This model is available in the NetLogo [27] model library and simulates the spread of fire through a forest represented by a 2D lattice where the fire is started on one side of the forest [25], see Fig. 3.1. The model contains only one parameter, the density of trees in the forest. How far the fire spreads depends on the density, but also has a dependence on the (randomly) initialised location of the trees. Here a fire dies out rapidly in sparse forests, or burns the majority of trees in a dense forest. In this model a fire can spread in any direction in the 2D lattice, but assumes there is no wind, so a fire can only spread from one tree to an unburned tree directly next to it.



Figure 3.1: NetLogo Interface for the Fire model. The only tunable parameter is forest density.

#### 3.1.2 The Model: Details

The density of the forest,  $\rho$ , is the only controllable parameter in this model. The level of noise in this model is low, however the number of trees burnt for a given forest density will vary between runs. This is due to the random initial configuration of the forest, i.e. where the trees are. We analyse the dependence of burnt trees in the forest with the initial density  $\rho$  of the forest. By considering the density of burnt trees ( $\kappa$ ) and the initial density ( $\rho$ ) in terms of percentage of the forest, we obtain a steady state equation in the system

$$F(\rho, \kappa^*) = \kappa^* . \tag{3.1}$$

In other words, for a given initial forest density,  $\rho$ , how much of the forest is burnt,  $\kappa$ . Also, how does this change with  $\rho$ ?

#### 3.1.3 Requirements of the User

We define the inputs in the same way as for the agent-based models (ABMs) in the other examples. As we have the microscopic description of this system, we can supply the values for the model parameters, here density  $\rho$  only, and use EMBER to obtain the macroscopic behaviour of this system, which is not obtainable with classical mathematical techniques. The model does not have any explicit output variables, however in this simple example we can see how these can be defined. The term 'output' in this context is vague, it is not necessarily the final metrics that the model produces. That is, the appropriate quantity may be a combination of, or scaled variable. In this example the model does not have an explicit value that is returned at the end of the simulation. However it is possible to obtain such a metric by defining our own metric. Intuitively here it is the percentage of burnt trees at the end of the simulation. Thus we can define our *Restrict Operator* as

$$\frac{burned - trees}{initial - trees}100 , \qquad (3.2)$$

where we factor in the initial number of trees to account for larger forests having more trees to burn. This is an unusual model in that is has no explicit output variables, but also no input variables. The only user controllable value is the density of the forest, the parameter that EMBER will analyse. As a consequence the *Lift Operator* in this case does nothing and therefore defining a 'dummy' variable *density-burnt* that doe not affect the model is sufficient. The user defined settings in the input file are given below.

```
String NetlogoFile = "netlogo/Fire.nlogo";
String[] systemParameters = {"set density"};
double[] param = {10.0};
String[] RestrictOperator = {"(burned-trees / initial-trees) * 100.0"};
String[] LiftOperator = {"set density"};
double[] Initial = {10.0};
boolean isSystemInitialised = false;
```

Here we vary the initial density of trees in the forest to see how this affects the number of burned after simulation.

A requirement of EMBER is to define the parameters; Realisations, Number of Steps, Time Horizon, Continuation Step and the direction of this step Continuation Step Sign. The Number of Steps is simply how many points you want to calculate, i.e. if you want to go from 0 to 1 in steps of 0.01 then this would be 100. In this example 0.01 would be the Continuation Step and the direction would be +. If you wanted to go from 1 to 0 then Continuation Step would still be 0.01 and Continuation Step Sign would be -. The Time Horizon is the length of the time simulation and Realisations is the number of independent simulations in the ensemble used by EMBER. The Number of Steps and Continuation Step Sign are in complete control of the user. The other parameters can be either be select using EMBERs configuration phase.

#### 3.1.4 Outcome of Equation-free Analysis

The results of the analysis are shown in Fig. 3.2. This illustrates the existence of a critical point with a nonlinear threshold, a common feature in complex systems [25]. Moreover this threshold is over a narrow range of forest density, which can be set between 0 and 100. There is a clear shift from a state of few burnt trees to one of almost complete destruction of the forest. A transition occurs between these two states over a narrow window of density ( $\rho$ ). From Fig. 3.2 it is also clear that the uncertainty changes during this transition, increasing to a maximum, then decreasing as the system reaches the other state. This is an important observation as it highlights a change in behaviour of both the model and its statistical behaviour. The random initial configuration of trees in the model causes there to be a distribution of simulation outcomes.

In fig. 3.3 it is clear that the distribution is not Gaussian (normally distributed) at any point. This highlights the need for robust statistical methods such as the bootstrap method [7], used here, which do not require assumptions of the underlying distribution. Interestingly we see a much larger variation in the amount of burnt forest when its density is around 60% giving rise to the larger error bars in Fig. 3.2 in this region.



Figure 3.2: Macroscopic behaviour of the forest fire model and its dependence on the initial density of trees.

#### 3.2 Ising model: a generalised transition model

#### 3.2.1 The Model: An Overview

This is a model of a magnet based on the magnetic moments (spins) of the atoms it contains [28] and is available in the NetLogo model library [27]. In this model each element of a 2D lattice represents an atom (agent) in the magnet with a spin either up (+1) or down (-1). Each spin state can be influenced by the temperature of the magnet and the magnetic moment of four neighbouring atoms. The overall spin of the magnet, known as the magnetization (M), is based on the sum of the spins of k atoms in the magnet,

$$M = \sum_{i=1}^{k} \delta_i, \quad ; \quad \delta_i = \begin{cases} 1 & : spin \ up \\ -1 & : spin \ down \end{cases}$$
(3.3)

#### 3.2.2 The Model: Details

The behaviour of the system is dependent on temperature T, in the low temperature limit the system is a ferromagnet and in the high temperature limit the system is in a paramagnetic state. The energy,  $\epsilon$ , of the atom at the i, j location is defined as the negative sum of the product of the spin of an atom and its four neighbours (nbh);

$$\epsilon_{i,j} = -\sum_{k=1}^{non} \delta_{i,j} \delta_{nbh} = -\delta_{i,j} \left( \delta_{i,j-1} + \delta_{i,j+1} + \delta_{i-1,j} + \delta_{i+1,j} \right) .$$
(3.4)

The energy is a measure of similarity or difference of an atom's spin state to that of its neighbouring atoms with a maximum value of 4, where an atom is the opposite to all of its neighbours, and a minimum of -4, where all five atoms have the same spin state. The probability, p, of an atom changing to the opposing spin state is determined through the Metropolis algorithm;

$$p = e^{-\Delta\epsilon/T} , \qquad (3.5)$$

where  $\Delta \epsilon$  is the difference in energy caused by an atom changing its spin state. That is the difference in Equation (3.4) between  $\delta_{i,j}$  and  $(-\delta_{i,j})$ . Equation (3.5) states that the *p* decreases for increasing  $\Delta \epsilon$ , but



Figure 3.3: Distributions of simulations for given values of forest density.

that p also increases with T. This example represents an abstract model that can simulate phase-transition behaviours in physical and social sciences [28]. Here we perform equation-free analysis by varying T and computing  $f(M^*;T) = M^*(T)$ , the results of the analysis are given in Fig. 3.4.

#### 3.2.3 Requirements of the User

We define the inputs in the same way as for the agent-based models (ABMs) in the other examples. As we have the microscopic description of this system, we can supply the values for the model parameters, here temperature T only, and use EMBER to obtain the macroscopic behaviour of this system, which is not obtainable with classical mathematical techniques. The model has only one variable, *magnetization*, the sum of the individual magnetic states in the systems. As there is only one variable in the system, our *Lift Operator* and *Restrict Operator* are simply defining and calculating the *magnetization* respectively.

However, the default model is available in the NetLogo model library, however can only initialise the state of the lattice with all atoms (in version 5.0.5); spin up (M=+1), spin down (M=-1), or randomly orientated (M=0). Thus we need to edit the *setup* procedure in order to be able to initialise the model to any *magnetization*, not just the three options in the default model.

```
to setup
  clear-all
  ask patches
   [ ifelse initial-magnetization = 0
      [ set spin one-of [-1 1] ]
      [ set spin initial-magnetization ]
      recolor ]
  set sum-of-spins sum [spin] of patches
```



Figure 3.4: Ising model interface with one controllable parameter.

```
reset-ticks end
```

In order to enable the appropriate macroscopic state to be translated to the microscopic simulations using the *Lift Operator*, a small modification is made to the **setup** procedure in the model. This modification (shown below) enables the magnetization M to be initialised at any value [-1:+1] by defining **initial**M, which is lifted from the macroscopic state.

```
to setup
  clear-all
  ask patches [ lift ] ;; new procedure in place of +/-1 or 0
  set sum-of-spins sum [spin] of patches
  reset-ticks
end
```

Here we also include a new procedure, Lift, to appropriately initialise the model.

```
to lift
  let ptype random-float 2.0
  let random-number ptype - 1.0
  ifelse initialM < random-number [ set spin -1 ] [ set spin 1 ]
  recolor
end</pre>
```

In newer versions of NetLogo, e.g. from 5.2.1, this step is not necessary as the setup procedure in the default model enables any magnetisation in the range M = [-1:+1].

```
to setup
  clear-all
  ask patches [
    ifelse random 100 < probability-of-spin-up
      [ set spin 1 ]
      [ set spin -1 ]
      recolor</pre>
```

```
set sum-of-spins sum [spin] of patches
reset-ticks
end
```

Note however, we include our amended *setup* procedure to illustrate how this may have to be edited to use EMBER as in general the default *setup* may not be sufficient.

The user defined settings in the input file are given below.

```
String NetlogoFile = "netlogo/Ising.nlogo";
String[] systemParameters = {"set temperature"};
double[] param = {2.5};
String[] RestrictOperator = {"magnetization"};
String[] LiftOperator = {"set magnetization"};
double[] Initial = {0.0};
boolean isSystemInitialised = false;
```

A requirement of EMBER is to define the parameters; Realisations, Number of Steps, Time Horizon, Continuation Step and the direction of this step Continuation Step Sign. The Number of Steps is simply how many points you want to calculate, i.e. if you want to go from 0 to 1 in steps of 0.01 then this would be 100. In this example 0.01 would be the Continuation Step and the direction would be +. If you wanted to go from 1 to 0 then Continuation Step would still be 0.01 and Continuation Step Sign would be -. The Time Horizon is the length of the time simulation and Realisations is the number of independent simulations in the ensemble used by EMBER. The Number of Steps and Continuation Step Sign are in complete control of the user. The other parameters can be either be select using EMBERs configuration phase.

#### 3.2.4 Outcome of Equation-free Analysis

Simulating the model reveals two district regimes depending on the temperature of the system, as illustrated in Fig. 3.5. Clearly the behaviour of the model is dependent on temperature, one question is how? Additional questions maybe; where is the transition between the regimes? What causes the multiple outcomes in the first regime? Is the transition smooth or abrupt? These questions can be answered using this method.



Figure 3.5: Typical outcome of Ising model simulations demonstrating the presents of two disctint regimes.

The first regime is known as the ferromagnetic state where the system exhibits spontaneous magnetisation and converges to either outcome in Fig. 3.5. Looking at the parameter dependence curve, Fig. 3.6, we can see that the initial conditions determine which of these states the system converges to. The blue line in Fig. 3.6 denotes an unstable state causes the transient behaviour in this model and denotes the approximate boundaries for convergence to either final state. In other words, if you are above the blue line you will tend to the M = 1 state, and the M = -1 state if you are below it. The location of the unstable state is important for denoting the basin of attraction of each state, potentially revealing an asymmetry in the system.

Also in this region, the systems is fixed in the  $\pm 1$  spin state until the temperature is above a threshold T = 1. Below this threshold the system is in a full state of lock-in. For temperatures above this threshold the system is still in a state of lock-in though it is possible for the alternative state to be sustained, i.e. not fully locked-in. At temperatures below T = 1 all the agent states are random thus there is no alignment. Although the probability according to Equation (3.5) is low in this phase, once an atom has flipped to the lower energy spin state it will extremely unlike to flip back resulting in a slow transition from M = 0 to

M = |1|. This explains the observation of transitions away from M=0 to one of the stable branches in this temperature region.

From Fig. 3.6 we see a rapid change in magnetization as T increases during the transitions from regime 1 to regime 2. This transition is rapid and can appear to be discontinuous around the critical point T = 2.3.

In region (2) of Fig. 3.6, the system is now in a paramagnetic state. Here the state of each agent is independent leading to an overall magnetization of zero due to random orientation of the elements. Regardless of the initial conditions, in this state the system will always converge to a magnetization of zero, as indicated by the lines in Fig. 3.6 (2).



Figure 3.6: Macroscopic behaviour of the Ising model and its dependence on temperature. Regions observed in Fig. 3.5 are also indicated here with arrows denoting the overall tranisent behaviour.

The number of ticks in Fig. 3.5 illustrates the slow dynamics of this model, particularly in the ferromagnetic state where convergence takes 10's of millions of time steps. This indicates the issue of analysis of ABMs with simulations alone, where a single simulation takes around 40s (running at maximum speed) for this number of ticks. Then when considering repeat simulations for averaging, and parameter variation, this can rapidly become a long computational process. The analysis used here does not require full length simulations while varying T, like other analysis tools, thus the overall computation time for analysis is significantly reduced. This becomes increasingly important for larger agent-based models (ABMs) where a single simulation can take a long time to run.

### Chapter 4

# Agent-based models: complex models

#### 4.1 A model of altruism

#### 4.1.1 The Model: An Overview

The Altruism model [26] simulates the competition between selfish and altruistic agents for survival in a 2D world. This model is available from the NetLogo model library [27]. In general the outcome of this battle depends on the environmental conditions (parameters) though there is a stochastic element coming from the initial configuration of the system, i.e. the randomised positioning of the agents. The system is randomly initialised where each space element in the 2D lattice has a defined probability of being either selfish, altruistic or empty. Selfish and altruistic agents compete for occupation of positions in the world based on simple rules but leading to some complex behaviour and nonlinear dynamics.



Figure 4.1: Altruism interface with several controllable parameter.

#### 4.1.2 The Model: Details

In this model agents are patches in the 2D lattice and compete for occupation of space in the world. In this model groups of five agents, a central agent and its surrounding neighbours, compete for occupation of the central patch. The world wraps around such that an agent on the 'edge' of the world have neighbours on

the opposite edge, thus there is a continuous flow of agents and neighbours. The occupation of the central location is based on the agents fitness values. All agents have a fitness value calculated as;

$$F_A^n = 1 - C + B \frac{N_A^n}{5} , (4.1)$$

$$F_S^n = 1 + B \frac{N_A^n}{5} , (4.2)$$

$$F_E^n = H . (4.3)$$

where  $F_A$ ,  $F_S$  and  $F_E$  are the fitness values for altruistic, selfish and empty agents respectively. Empty agents correspond to locations where there is neither an altruist or selfish agent. System parameters  $C_A$ ,  $B_A$ and H represent the cost and benefits of altruism, and the harshness of the environment respectively. The term  $\sum_{n=1}^{nbh} N_A/5$  is the average number of altruists in the neighbourhood (*nbh*), that is the central location and its four neighbours. At each unit of time an agent and its four neighbours use their fitness values to compete for occupation of the central location by using weighted probability, P, calculated as:

$$p_{A_{i,j}}^{n+1} = \frac{N_A^n F_A^n}{F_T^n}$$
(4.4)

$$p_{S_{i,j}}^{n+1} = \frac{N_S^n F_S^n}{F_T^n} \tag{4.5}$$

$$p_{E_{i,j}}^{n+1} = \frac{N_E^n F_E^n + D}{F_T^n}, (4.6)$$

where  $N_Z^n$  is the number of agents of type Z in the neighbourhood of (i, j), i.e.  $(i \pm 1, j \pm 1)$ ,  $F_A^n, F_S^n$  and  $F_E^n$  represent the 'fitness' of each agent for a given type. where

$$F_T^n = N_A^n F_A^n + N_S^n F_S^n + N_E^n F_E^n + D , (4.7)$$

Here  $N_i$  is the number of *i*-type agents and D is the rate of disease in the environment. Agents enter a lottery for the central position based on these seeds. A random number, r, is generated between 0 and 1, and the central position becomes; an altruist if  $r < P_A$ , selfish if  $r < P_A + P_S$ , and empty if  $r > P_A + P_S$ . This process continues at each time iteration for all locations simultaneously, which with the random initial configuration of agent location, leads to stochastic behaviour. There is a conservation in the model with the total number of patches, where patches can be either either Altruists or Selfish agents, or Voids. The relative number of each patch type must sum to unity,  $N_A + N_S + N_V = 1$ , therefore the number of agents under any settings is confined to a plane in  $g(N_A, N_S, N_V)$  space.

#### 4.1.3 Requirements of the User

We define the inputs in the same way as for the agent-based models (ABMs) in the other examples. As we have the microscopic description of this system, we can supply the values for the model parameters. The parameters in this model are the rate of *disease*, the *cost-of-altruism*, *benefits-of-altruism* and the *harshness* of the environment. He we are interested in how this macroscopic behaviour of this model varies with the rate of *disease* in the system. The mode has three types of agents, and thus a simple output measure for our *Restrict Operator* is the number of agents after the simulation. However, as this model has a finite total number of agents, determined by the size of the lattice in the model (which can be defined by the user), we can obtain all three values from only calculating the number of two types of agents. That is

$$N_A + N_S + N_E = N_T, (4.8)$$

and as  $N_T$  is fixed during the computation, if we calculate the number of altruists and selfish agents as ratios of the total number of agents we obtain two measures that full determine the system as

$$\frac{N_A}{N_T} + \frac{N_S}{N_T} = 1 - \frac{N_E}{N_T} \ . \tag{4.9}$$

Thus our *Restrict Operator*, in NetLogo syntax, is

count	patches	with	[pcolor =	pink ] /	/ cou
unt	patches	with	[pcolor =	green]	/ cou

where altruists and selfish agents are coloured pink and green respectively. Similarly, the *Lift Operator* is simply defining number of altruistic and selfish agents in this way. The user defined settings in the input file are given below.

```
String NetlogoFile = "netlogo/Altruism.nlogo";
String[] systemParameters = {"set disease ", "set cost-of-altruism ", "set benefit↔
  -from-altruism ", "set harshness "};
double[] param = {0.4, 0.13, 0.48, 0.85};
String[] RestrictOperator = {"count patches with [pcolor = pink] / count patches "↔
  , "count patches with [pcolor = green] / count patches "};
String[] LiftOperator = {"set altruistic-probability ", "set selfish-probability ↔
  "};
double[]xInitial = {0.5, 0.5};
boolean isSystemInitialised = false;
```

A requirement of EMBER is to define the parameters; Realisations, Number of Steps, Time Horizon, Continuation Step and the direction of this step Continuation Step Sign. The Number of Steps is simply how many points you want to calculate, i.e. if you want to go from 0 to 1 in steps of 0.01 then this would be 100. In this example 0.01 would be the Continuation Step and the direction would be +. If you wanted to go from 1 to 0 then Continuation Step would still be 0.01 and Continuation Step Sign would be -. The Time Horizon is the length of the time simulation and Realisations is the number of independent simulations in the ensemble used by EMBER. The Number of Steps and Continuation Step Sign are in complete control of the user. The other parameters can be either be select using EMBERs configuration phase.

#### 4.1.4 Outcome of Equation-free Analysis

In this model we continue in the disease parameters D for several system configuration; 1)  $N_S = 0$  (no selfish agents), 2)  $N_A = 0$  (no altruistic agents) and 3)  $N_S = 0$  and  $N_A = 0$  (no agents) given in Fig. 4.2. This illustrates the different dependence on D with Altruistic and Selfish agents varying nonlinearly and linearly respectively. Additionally we can see that the two agents differ in stability, altruistic branch beginning unstable and the selfish branch beginning stable.

The critical points in Fig. 4.2 show the empty branch inheriting the stability of the other curves. The empty branch collides with the unstable Selfish branch on the boundary of regions 3 and 4, causing the empty state to become stable and giving rise to the the birth of an unstable altruistic state. This new branch remains unstable until the boundary of region 4 and 5 where it collides with the stable Altruist branch resulting in a tipping point. Here the stable altruist state appears to vanishes just above disease values of 0.6. Beyond this tipping point, only a single stable state exists in the system with no Altruistic or Selfish agents.

Additionally the Altruistic and Selfish branches change stability in a narrow window, but at different values of Disease. In this window (region 2) there is a transition where the system moves between the  $N_k = 0$  and k = A, S planes. That is there is a branch between the Altruistic and Selfish curves in Fig. 4.2 of fixed points where both Altruistic and Selfish agents exist. Note due to the conservation of the number of agents, this branch cannot be drawn on Fig. 4.2 directly. This connecting branch is highly unstable and rapidly converges to one of the two agent branches, which are both stable in region 2.

The transient behaviour of the system can be better understood using Schematic diagrams of the stability in each region. Figure 4.3 represents the plane in  $g(N_A, N_S, N_V)$  space where the number of patches is conserved and are qualitative representations of the dynamics in the model for each region. Moreover, these diagrams also provide a clear description of the transitions occurring on the boundary of each region. In region 1 there are two unstable states (one doubly unstable at the origin) and a stable state consisting of only selfish agents and voids. Qualitatively we can see that system will always tend to a state where there are no Altruists, that is, the selfish agents always win out. Note that the case where D = 0 behaves the same as in region 1, however the stable point along the  $N_S$  axis and the unstable point along the  $N_A$  axis are both at 1. That is in the special case where there is no Disease, agents do not die and will always converge to a full population of one agents. As D is increased further the system undergoes a bifurcation as it enters region 2, with the unstable altruistic branch becoming stable and giving rise to the creation of an unstable state in between the stable altruistic and selfish branches. This fits directly with simulations in this region, where the system can converge to either of the stable branches, meaning the outcome of the battle between the two agents dependent on the initial conditions of the system. We note the lower simulation results in region 2 of Fig, 4.3 where the system initially converges to the unstable state where both altruists and selfish agents exist before converging on to the stable altruist branch. Due to the level of noise in the model, and the low occurrence of convergence to this unstable state, this observation would not be possible without the



Figure 4.2: Parameter dependence curve for the altruism model. Stability is indicated by solid (dashed) lines for stable (unstable) branches and numbers indicate regions of behaviour separated by critical point. Note region 2 occurs at a single Disease value.

knowledge of fixed points and stability obtained through the EF continuation here. Increasing D further causes this new unstable state to collide with the stable state on the selfish branch making it unstable in region 3. This is an important parameter regime in the context of this model, as it is the first time that the altruists agents become 'favourite' to win out over the selfish agents. From Fig. 4.3 we can see that behaviour in region 3 is similar to region 1 with the altruist and selfish branches having changed stability. At the next critical point, the boundary of regions 3 and 4, the now unstable selfish state collides with the void state, leaving the void branch stable and creating an unstable altruist branch. In this region will always tend to a state with no selfish agents, and the fate of the altruist agents is dependent on the initial conditions. If the initial number of agents in above the dashed line in Fig. 4.2 in region (4), the system will converge to the solid line and the population of altruists is sustainable. However, if the initial number of altruists is below this dashed line, then the population is not sustainable and all agents will die out. In this case the system will converge to the solid black line (empty state). The final transition occurs when the only reaming stable (on the altruist Branch) collides with the new unstable altruist state yielding only one, stable, state of the system. Regardless of initial conditions, in this region all agents will die leaving the world full of voids, which you can observe directly with Disease settings in the region.



Figure 4.3: Stability and transient behaviour of altruism model in the regions in Fig. 4.2. Agents are indicated by pink (altruists), green (selfish) and black (empty, no agent). Circles represent stable (black) and unstable (white) solutions and arrows indicate the trajectory of the dynamic behaviour. Note the origin of these plots indicate a void only state. Below each region is an example of the model behaviour under this configuration, all of which are initialised from the same settings (top), except for the value of D. In each case running simulations only will always converge to the stable states in regions (bottom row).

### Chapter 5

# **Additional Information**

#### 5.1 Publications

All the work in this guide is for demonstration purposes and is not to be referenced in any formal documents. In all examples citations to the origin of the models are included in the relevant sections. Users who wish to refer to any of these models should cite the original publication of the model, whether it is open source or not. For citations to the analysis of the model contained in this guide users, readers should refer to the published works given in the section or to [23]. All published examples here are explicitly labelled in the sections where appropriate. If you wish to reference the analysis of a model in this guide that is not published, or not yet published, please see §1.8 for the relevant contact information. This guide is designed to be informative, but not exhaustive. New applications, analysis and features are being added dynamically and should be added to this guide where possible to help other users. External users who would like to add to this guide with their models please contact the team using the information in §1.8.

A list of published, submitted and work in preparation:

- Equation-free analysis of agent-based models and systematic parameter determination, S. A. Thomas, D. J.B. Lloyd, and A. C. Skeldon, Physica A, 2016 http://personal.maths.surrey.ac.uk/st/D.J. Lloyd/EFNCpaper.pdf
- Analyzing regime shifts in agent-based models with equation-free analysis, R. Martin and S. A. Thomas, International Environmental Modelling and Software Society (iEMSs) 8th International Congress on Environmental Modelling and Software, 2016 http://www.iemss.org/society/index.php/iemss-2016-proceedings
- A systematic review of altruism, S. A. Thomas, 2018 (in preparation)
- EMBER: a macroscopic analysis tool for agent-based models, S. A. Thomas, 2018 (in preparation)

#### 5.2 Future work

The development of a GUI for the code will be a focus of future efforts.

Algorithmic improvements including GMRES methods for determining stability of solutions as theses have been demonstrated to be more robust, the continuation of periodic solutions to increase generality of the algorithm and the extension to parallel processing to reduce computational run times of the code.

Beyond algorithmic developments, further application of EMBER to models is an ongoing project. This provides insight in to the application and can also highlight additional elements to include in future versions of the code.

#### 5.3 External algorithmic developments

The object oriented structure of EMBER allows simple updates or replacement of algorithms without affecting the underlying processes of the code. Each mathematical or computational process in EMBER is encapsulated so can be edited or replaced directly with minor changes to the relevant interfaces. It is advised that only those comfortable with Java attempt this as there will be no support for third party changes to the source code. Additional features if generally useful, more accurate or robust than those currently in EMBER are welcome and users should see contact details in §1.8 for these purposes.

### 5.4 Further Reading

The following table contains a reference list for further reading on the topic contained in this method and example.

Topic	Reference
Introduction to bifurcation analysis	[16]
Introduction to continuation	[1,  6,  13,  20]
Introduction to equation-free methods	[11, 12, 22, 23]
Further models for altruism	[3, 4, 8-10, 17-19, 21]
Double well potential	[2, 5, 14, 15, 24]

### Glossary

agent an actor or individual entity. 14

Altruism an action(s) that benefit others that may come at a cost to the individual or group. 14

- **basin of attraction** boundary of attraction for a stable state, typically denoted by the presences of an unstable state. 13
- **bootstrap method** a method to calculate the statistics, e.g. mean and variance, of any distribution. The technique approximates these values through random sampling with replacements and provides a robust method without knowing the underlying distribution or equations to calculate the statistics . 9
- complex behaviour non-trivial behaviour. 14
- critical point a point of change in a systems or its behaviour, such as at a particular parameter value. 5, 9, 16, 23

driver a parameter in the system that causes a change to the output of the system, or its behaviour. 5

dynamics how a model or system behaves over time. 14

- encapsulate a computer science term often used in object oriented programs, where algorithms are combined with the data structures they act on to enable the separation of process in a code leading to a modular design. 25
- equation-free a technique where the mathematical description of a system or model can be replaces by an appropriately initialised ensemble of microscopic simulations . 5
- **ferromagnet** state of a material where all the individual elements, which each carry a charge, tend to align with each other resulting in the material *spontaneously* developing a magnetic charge determined by its initial state . 10, 12
- fitness measure of how good a strategy, agent, or entity is. 14
- **Gaussian** a symmetric distribution characterised by a mean and standard deviation, also known as a normal distribution or bell curve. 9, 23
- **lock-in** where a system is trap in a given state, the dynamics of the system are independent of the parametrisation and completely determined by the initial conditions. 13
- magnetization the average magnetic state of a material consisting of a collection of individual magnetic elements or particles. 10, 13
- Metropolis algorithm a probabilistic algorithm dependent on energy and temperature. 11

**NetLogo** an agent based modelling language. 8, 10, 14

**object oriented** a computer science term for the structure of a program where abstract components, 'objects', are used in place of specific implementation of variables and data structures to improve generality and reuse of code . 21, 25

**paramagnet** state of a material where all the individual elements, which each carry a charge, are independent of each other, leading to an overall charge of zero . 10, 13

path-following tracing out lines in the system that represent stabilities, equilibriums or solutions. 5

phase-transition the change from one state to another. 11

- **regime** region in model space where the dynamics of the system are, qualitatively of quantitatively, the same. A transition occurs between regimes and may be continuous of discontinuous. 5, 12, 13
- **spin state** the state of an element or atom, with possible states being spin up (positive charge), or spin down (negative charge). 10

stability the ability for a state to exist in the presents of perturbations (stable), or diverge (unstable). 16

stochastic unpredictability due to random process in a system, model or equation. 5, 14, 22, 23

tipping point where a stable state appears to vanish at a critical point. 5, 17, 23

transient intermediate state of a system while its converges to a final state. 17

variance a statistical measure of the variability of a results over a number of repeated trials. 9

# Acronyms

ABM agent-based model. 5
ABMs agent-based models. 5, 6, 13, 21, 23
API application program interface. 21
EF Equation-free. 23
EF equation-free. 11, 12
EMBER Emergent and Macroscopic Behavioural ExtRaction. 5, 6, 21, 23, 25
ODE ordinary differential equation. 23

Acronyms

# Bibliography

- E. L. Allgower and K. Georg. Numerical Continuation Methods, An Introduction. Springer-Verlag Berlin Heidelberg, 1990.
- [2] D. Barkely, I. G. Keverekidis, and A. M. Stuart. The moment map: Nonlinear dynamics of density evolution via a few moments. SIAM J. Applied Dynamical Systems, 5:403–434, 2006.
- [3] R. Boyd, H. Gintis, S. Bowles, and P. J. Richerson. The evolution of altruistic punishment. Proceedings of the National Academy of Sciences, 100(6):3531–3535, 2003.
- [4] C. F. Camerer. Behavioral game theory: Experiments in strategic interaction. Russel Sage Fondation, New York, NY, 2003.
- [5] P. D. Ditlevsen and S. J. Johnsen. Tipping points: Early warning and wishful thinking. *Geophysical Research Letters*, 37(19):1–4, 2010.
- [6] E. Doedel, H. B. Keller, and J. P. Kerenevez. Numerical analysis and control of bifurcation problems (i) bifurcation in finite dimensions. *International Journal of Bifurcation and Chaos*, 1(3):493–520, 1991.
- [7] B. Efron and R. J. Tibshirani. An introduction to the bootstrap. New York: Chapman & Hall, 1993.
- [8] M. A. Janssen. The effect of social preferences on the evolution of cooperation in public good games. Advances in Complex Systems, 17(03n04):1450015, 2014.
- M. A. Janssen and C. Bushman. Evolution of cooperation and altruistic punishment when retaliation is possible. *Journal of Theoretical Biology*, 254(3):541–545, 2008.
- [10] M. A. Janssen and E. Ostrom. Vulnerability of social norms to incomplete information. The Complexity of Social Norms, pages 161–173, 2014.
- [11] I. G. Kevrekidis, C. W. Gear, J. M. Hyman, P. G. Kevrekidis, O. Runborg, and C. Theodoropoulos. Equation-free, coarse-grained multiscale computation: enabling microscopic simulators to perform system-level analysis. *Communications in Mathematical Sciences*, 1(4):715–762, 2003.
- [12] I. G. Kevrekidis and G. Samaey. Equation-free multiscale computation: Algorithms and applications. Annual Review of Physical Chemistry, 60(1):321–344, 2009.
- [13] B. Krauskopf, H. M. Osinga, and J. Galàn-Vioque, editors. Numerical Continuation Methods for Dynamical Systems. Springer, 2007.
- [14] C. Kuehn. Deterministic continuation of stochastic metastable equilibria via lyapunov equations and ellipsoids. SIAM Journal on Scientific Computing, 34:A1635–A1658, 2012.
- [15] M. C. Mahato and A. M. Jaynnavar. Some stochastic phenomena in a driven double-well system. *Physica A*, 248:138–154, 1998.
- [16] C. Meunier and A. D. Verga. Noise and bifurcations. J. Stat. Phys., 50(1-2):345–375, 1988.
- [17] M. Mohtashemi and L. Mui. Evolution of indirect reciprocity by social information: the role of trust and reputation in evolution of altruism. *Journal of Theoretical Biology*, 223(4):523–531, 2003.
- [18] M. A. Nowak and K. Sigmund. Evolution of indirect reciprocity by image scoring. *Nature*, 393:573–577, 1998.
- [19] J. W. Pepper. Simple models of assortment through environmental feedback. Artificial Life, 13(1):1–9, 2007.

- [20] W. C. Rheinboldt. Numerical continuation methods: a perspective. Journal of Computational and Applied Mathematics, 124:229–244, 2000.
- [21] M. Hartshorn T. Shultz and A. Kaznatcheev. Why is ethnocentrism more common than humanitarianism? pages 2100–2105. 2009.
- [22] C. Theodoropoulos, Y. H. Qian, and I. G. Kevrekidis. Coarse stability and bifurcation analysis using time-steppers: a reaction-diffusion example. Proc. Natl. Acad. Sci., 97:9840–9845, 2000.
- [23] Spencer Angus Thomas, Anne C. Skeldon, and David B. J. Lloyd. Equation-free analysis of agent-based models and systematic parameter determination. *Physica A*, 1 (1) May:1–42, 2016.
- [24] G. Q. Cai W. Q. Zhu and R. C. Hu. Stochastic analysis of dynamical system with double-well potential. Int. J. Dynam. Control, 1:12–19, 2013.
- [25] U. Wilensky. Netlogo fire model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, http://ccl.northwestern.edu/netlogo/models/Fire, 1997.
- [26] U. Wilensky. Netlogo altruism model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, http://ccl.northwestern.edu/netlogo/models/Altruism, 1998.
- [27] U. Wilensky. Netlogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, http://ccl.northwestern.edu/netlogo/, 1999.
- [28] U. Wilensky. Netlogo ising model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, http://ccl.northwestern.edu/netlogo/models/Ising, 2003.