EMBER: Emergence and Macroscopic Behavioural ExtRaction

Spencer A. Thomas *1 , Anne Skeldon $^{\dagger 2}$, and David B. J. Lloyd $^{\ddagger 2}$

¹National Physical Laboratory, Teddingon, TW11 0LA, UK ²Department of Mathematics, University of Surrey, Guildford, GU2 7XH, UK

May 14, 2018

1 Program Overview

EMBER (Emergence and Macroscopic Behavioural ExtRaction is a generalised analysis tool for numerical continuation and bifurcation analysis of stochastic systems or those without a closed form solution or macroscopic equation of state. The code is based on equation-free techniques to *side step* the need for these equations and thus we can perform continuation, bifurcation analysis and extract statistical descriptors of a system or simulation model. EMBER also contains a method for determination of necessary equation-free parameters based on the dynamics of the system and a robust continuation method to coping with highly noisy systems. The algorithms behind the EMBER code are introduced in detail in this paper [1].

EMBER is written in java for compatibility with windows, mac and linux operating system, object-orientated design (OOD) and also for the flexibility with API to external models. The OOD enables aspects of the code to be encapsulated to increase code usability and upgrades. For instance, the bifurcation detection is based on the change in sign of the determinate of the Jacobian, a simple but un-robust method, which is defined in an isolated module of the code and could be substituted for another method such as GMRES without affecting the rest of the code. Another example is the API which is the only part of the code that interacts with the external model, therefore any system can be analysed with this code providing a API can be written. Currently this program will analyse java based simulations and agent-based models written in Netlogo, further developments will extend this to other systems and languages.

2 Author Synopsis

ST was the lead developer on this project during a postdoc at the Department of Mathematics, University of Surrey as part of the Evolution and Resilience of Industrial Ecosystems (ERIE) project through the support of the UK Engineering and Physical Sciences Research Council for programme grant EP/H021779/1. ST is now a Higher Research Scientist (PostDoc) at the National Physical Laboratory.

AS and DL are both readers in Mathematics at the University of Surrey and were a co-Investigators throughout the ERIE project. Both AS and DL defined the initial problem statement and supervised the project.

All three authors contributed to the development of the program with ST developing the java based code and underlying algorithms with the support of AS and DJ.

^{*}spencer.thomas@npl.co.uk; spencer.angus.thomas@gmail.com

[†]a.skeldon@surrey.ac.uk

[‡]d.lloyde@aurrey.ac.uk

3 Material Content

The material contains the compiled and source code for users to use and modify the code. There is also a user guide with some example systems and tips for users. This is recommend for users who are reasonably comfortable with running scripts and a basic knowledge of prgram

4 Instructions for Users

4.1 Download and instillation procedure

The current version of the code is available for down load from https://bit.ly/2HRJ4qV or http: //epubs.surrey.ac.uk/840475/ which contains all the required directories and sub folders. In order to run the code, Java 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.

4.2 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 sin

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.

4.3 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 EMERB's 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 [1] 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.

5 Model Definition

All models, or APIs to external simulators are defined in MarcoOperator.java. For simplicity we include two version of the code. Firstly for stochastic equations (defined in codes/MacroOperator.java with the stochastic micro models) which can be run using the ./compile.sh and ./run.sh scripts. The second version is included for accessing external models or simulators in place of an analytical description of the system. This is achieved in EMBER via an API defined in codesABM/



Figure 1: Continuation of a stochastic double well potential is compared to the deterministic case (black line) with where stable (unstable) branches are illustrated by red (blue) points.

MarcoOperator.java. The code for EMBER is identical in *codes*/and codesABM with the exception of the *MarcoOperator.java* code which includes an API in place of the equation in the first version.

6 Example Systems

Here is a collection of a few illustrative systems that have been analysed with EMBER. Firstly we consider the stochastic double well system defined as

$$dx = -\frac{dV(x)}{dx}dt + \eta \, dW \;, \tag{1}$$

where η is the noise level and W is Gaussian noise with mean zero and variance one, and

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

where μ is the barrier height and ν is the asymmetry of the well about x = 0. The potential V is drawn in Fig. 1 for fixed ν and three different values of μ .

EMEBER is highly robust due to an algorithm for restricting convergence in the continuation as shown in Fig. 2.

We can also analyse agent-based models without requiring an analytical description of the model. Here we consider a simple 1-D Ising model implemented here http://ccl.northwestern.edu/netlogo/models/Ising. EMBER is able to extract the dynamical behaviour of the system revealing a pitch-fork bifurcation as shown in Fig. 3.

Finally we consider a model of altruism implemented as an agent-based model here http://ccl. northwestern.edu/netlogo/models/Altruism. Here altruistic and selfish populations complete for survival on a 2D lattice. The depth of analysis EMBER can uncover from a system is illustrated in Fig. 4.

References

[1] 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.



Figure 2: A convergence restriction (C^3R) in EMBER enabling very robust continuation at high levels of noise for a low number of realisations for both secant (SC) and least squares (LS) based predictors.



Figure 3: A pitch-fork bifurcation reavealed from analysis of the Ising agent-based model.



(b) Bifurcation diagram

Figure 4: Rich dynamical information extracted from the sysmte using EMBER.