UTRECHT UNIVERSITY

MASTER THESIS

Homeostatic behaviour in democratic societies under climate change

Author:
Bruno RODRIGUES

Supervisors:
Prof. dr. ir. Henk STOOF
Prof. dr. ir. Henk Dijkstra

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Complex Systems Studies
Institute for Theoretical Physics

August 4, 2020

"We don't have time to sit on our hands as our planet burns. For young people, climate change is bigger than election or re-election. It's life or death."

Alexandria Ocasio-Cortez

UTRECHT UNIVERSITY

Abstract

Faculty of Science
Institute for Theoretical Physics

Master of Science

Homeostatic behaviour in democratic societies under climate change

by Bruno RODRIGUES

Homeostatic behavior is a regulative mechanism in complex systems where internal state variables are maintained within certain bounds under varying external conditions. In this project we investigate how different parameters lead to regimes of relative stability in election outcomes and under which circumstances landslides occur in democratic societies under climate change. We make use of DICE, an integrated assessment model of the world's climate-economy, and an Ising-like opinion dynamics model on a 2D-lattice small-scale network which allows for external stimulation. The topology of the network is shown to have an impact on election results as well as affecting its statistical properties. The effects of authority, connectivity and clustering between individuals play a heavy role in opinion formation and affect its critical points between ferromagnetic and paramagnetic phases. A coupling mechanism between the climate-economy feedbacks and their effect on voting behaviour was formulated. The coupling involves the optimization of an abatement policy to mitigate climate-induced damages to the world economy by maximizing total welfare, which is distinct for each political party. A projection of election outcomes (and their consequences to the climate-economy) is then numerically estimated by making use of Monte Carlo simulations. We show that the main driving factor in voter perception is the tax that results from abatement policy and that this leads to an upper and lower bound to external stimulation. The climate-induced damages play a minimal role in the short term, but are shown to take over in later years and force the voters into a ferromagnetic state.

Acknowledgements

I'd like to thank those that are part of the collective academic "we" that were invaluable for the completion of this thesis and are an integral part of its final outcome.
Firstly, a big thank you to both my thesis supervisors Prof. Dr. Henk Stoof and Prof. Dr. Henk Dijkstra. Both provided me with the guidance I required from their respective fields of expertise and the inspiration to tackle such a new an unconventional topic. To Prof. Stoof, a special thanks for daring to broaden the scope of physics research into new and challenging interdisciplinary fields; and to Prof. Dijkstra, a special thanks for reinvigorating my passion for science with such a bold topic.
I'd to thank Claudia Wieners for her enthusiasm for the topic and for all her suggestions on building the coupling equation. Her knowledge of DICE and its inner workings proved to be essential.
I want also to acknowledge Paolo Accordini for his assistance with all the technical nuances of Python.
On a more personal note, I would also like to thank the support of my friends; particularly Menno Kramer, Emmi Määttänen, Jorgos Papadomanolakis and Chiara Raffaelli for not only encouraging me but also listening to my ramblings regarding my thesis despite sometimes not understanding what I was saying, yet never dismissing me.

Contents

Dedicated to Paula and Gil.

Chapter 1

Introduction

In his *Theogeny* (c. 700 BC), the ancient Greek poet Hesiod narrates the story of the genesis of the deities of Greek mythology. After invoking the Muses, he tells of the generation of the first primordial deities:

> *"Verily at the first Chaos came to be, but next wide-bosomed Gaia (Earth), the ever-sure foundations of all the deathless ones who hold the peaks of snowy Olympus..."*[1]

Gaia is the ancestral mother of all life. She is the mother of Uranus (the Sky), from whose union she bore the Titans (themselves parents of many of the Olympian gods), the Cyclopes, and the Giants. This personification of Earth as a living being, an integrated whole, a deity, has a long tradition. Western culture colloquially refers to it as "Mother Nature". Gaia and the concept "she" represents has entered the scientific literature through the works of chemist James Lovelock and (co-developed) by the microbiologist Lynn Margulis in the 1970s [1, 2].

The Gaia hypothesis proposes that all organisms and their inorganic surroundings on Earth are closely integrated to form a single and self-regulating complex system, maintaining the conditions for life on the planet. This involves the biosphere, the atmosphere, the hydrospheres and the pedosphere, all tightly coupled as a single an evolving system. The system as a whole, which they call Gaia, seeks a physical and chemical environment optimal for the existence of contemporary life on Earth.

The scientific investigation of the Gaia hypothesis focuses on observing how the biosphere and the evolution of life forms contribute to the stability of global temperature, ocean salinity, oxygen in the atmosphere and other factors important for habitability in a preferred homeostasis[2].

Criticism of the Gaia hypothesis has been primarily focused on it being teleological — a belief that things are purposeful and aimed towards a goal. This description of the living Earth system as an entity with a conscious goal in mind is certainly unappealing and should be taken as merely a metaphorical description of Earth processes. Lovelock himself argues that no one mechanism is responsible for homeostasis, that the connections between the various known mechanisms may never be known or fully understood, and that this is accepted in other fields of biology and ecology as a matter of course.

Nevertheless, many processes in the Earth system are on a delicate balance and happen to somehow maintain the correct range of conditions required for complex life to exist. One notoriously important parameter is the average atmospheric temperature, which is critically regulated by the carbon dioxide (CO_2) content in the

[1] *"The Theogony of Hesiod"* (ll. 116-138). (https://www.sacred-texts.com/cla/hesiod/theogony.htm)

[2] The tendency towards a relatively stable equilibrium between interdependent elements, especially as maintained by physiological processes.

atmosphere. The greenhouse effect is well known to most literate adults, including the consequences of a greenhouse effect gone haywire: Venus. Since the mid-20th century these scientific concepts have seeped into the global vernacular as global warming has been accelerating due to anthropogenic influence. Global warming and its effects on the stability of the climate system are known as climate change.

The Intergovernmental Panel on Climate Change (IPCC) concluded that "human influence on climate has been the dominant cause of observed warming since the mid-20th century". These findings have been recognized by the national science academies of major nations and are not disputed by any scientific body of national or international standing [3, 4]. The largest human influence has been the emission of greenhouse gases, mostly from CO_2 and methane. Fossil fuel burning is the principal source of these gases, with agricultural emissions and deforestation also playing significant roles. Temperature rise is enhanced by self-reinforcing climate feedbacks, such as loss of snow cover, increased water vapour, and melting permafrost.

Climate change is therefore seen as a market failure by environmental economists. As human activity and respective economies continue to drive our energy needs upward, collective emissions of CO_2 through the burning of fossil fuels show no indications of slowing down. This has since become the defining political, social and ethical problem of the 21st century.

Previous attempts at modelling the feedbacks between the climate system and human economies have resulted in a variety of models, that inform on climate change policy (with somewhat questionable reliability). In this thesis, we are interested in adding the element of electoral outcomes to one such model, by coupling a social network of a democratic society to it and developing a mechanism that conveys the effects in an accurate fashion. By doing this, we want to explore the consequences that man made actions, through the economy and its political decisions, have on regulating the average world temperature and see if homeostasis occurs and under which conditions.

The structure of this thesis is as follows: In chapter 2, we introduce the basic concepts that are needed to understand our approach and methodology. We start with a brief explanation of the Ising model and its essential properties. An explanation of Markov chains and acceptance rates then follows suit, as this methodology is extended in the custom algorithm of Monte Carlo simulations that we employ to obtain our results. Lastly, a short overview of the fundamental concepts of graph theory that we will be repeatedly addressing is presented.

In chapter 3, we present the opinion dynamics model that we have chosen in order to tackle our problem as well as the motivations behind this particular choice, considering the extensive range of voter models that exist in the literature. We showcase the network structure, its topological properties and a description of how the algorithm for the network formation is implemented. The dynamics of opinion formation and the equations that govern it are explained, with the analogous behaviour to the Ising model being evident here. A detailed discussion of the algorithm and its capacity to generate networks with the properties required to emulate real networks is presented. Also, we show how opinion formation is influenced by multiple network parameters, network temperature and, most importantly, external stimulation, leading to phase transitions.

Chapter 4 is a short overview of the DICE model, which is how we have chosen to model the effects of climate chance. The equations of the model are explained as well as all the necessary concepts from both climate physics and economics. We abstain from lengthy discussions on the nuances involved, but particular attention is given to the core concepts that will be used in our extension of this model, such

as climate damages, climate sensitivity, average atmospheric temperature anomaly and abatement (cost of climate related policies). We end this chapter with a short discussion on multiple scenarios that can be model with DICE and its implications, which will then be used to compare with our results.

In chapter 5, we tackle the main problem we have set out to do by introducing a framework for coupling the DICE model with our opinion dynamics model. A brief description of this coupling and how we implement it for a two-party system is shown, as well as a discussion on the criteria we use that distinguishes the political parties from one another[3]. We then discuss our proposed coupling equation and the motivations behind its mathematical form, as well as its behavioural properties. Results are then showcased for a variety of parameters in order to explore the regimes under which our coupled DICE model operates with reasonably realistic outcomes. These results are obtained through Monte Carlos simulations and optimization algorithms, and we discuss the behaviour and implications of the model with particular attention to the impact of climate damages, network temperature and the importance of considering the future consequences of climate change into mediating opinion formation. Finally, we see how these different regimes and subsequent changes in electoral outcomes influence the atmospheric temperature anomaly curve and how it effectively bounds it to a range of possible scenarios.

In the appendices, we present a simplified version of the Python code used to perform our simulations, for the purposes of reproducibility and further development.

[3] Attempts at differentiating the political parties by asking them directly were made, as we've sent enquiries to multiple parties present in the Dutch parliament. However, of the responses we got, it seems that political party officials do not have a stance or knowledge on the climate change parameters/concepts we use in the fields of climate change economics.

Chapter 2

Theoretical Background

2.1 The Ising Model

In statistical physics, there is hardly a doubt that more hours have been spent thoroughly researching the properties of the Ising model than any other. An exact solution of its properties in three dimensions continues to elude us, despite countless and ever more sophisticated attempts at doing so. Nevertheless, a great deal about it is known from computer simulations, and also from approximate methods such as series expansions and perturbation theory.

The Ising model is a model of a magnet, named after Ernst Ising [5]. The basic assumption is that the magnetism of a bulk material is made up of the combined magnetic dipole moments of the many atomic spins within the material. The model postulates a lattice (which can be of any geometry we choose — the simple cubic lattice in three dimensions is a common choice in solid state physics; or simply a two dimensional grid as used in sociophysics' networks) with a magnetic dipole or spin for each atom.

The spins can then be described as a scalar s_i in binary form as being either 1 or -1, respectively corresponding to either "up" or "down" dipole states. Should enough spins align in a certain direction, it leads to the generation of a macroscopic magnetic field as described by classical electrodynamics. These interactions between neighbouring spins can also lead to an interesting phenomena; if enough spins are aligned in a certain direction, all spins in the lattice will align themselves in the same direction - this is called a phase transition.

In its simplest case, all interactions have the same strength, denoted by J which has dimensions of energy, and the interactions occur solemnly between spins of sites which are nearest neighbours on the given lattice configuration. An external magnetic field B that acts on the spins can also be added.

In this case the Hamiltonian H is;

$$H = -J \sum_{\langle i,j \rangle} s_i s_j - B \sum_{i}^{N} s_i \tag{2.1}$$

where the notation $\langle i, j \rangle$ represents the sum over nearest neighbours and N is the number of sites on the lattice.. Positive values of J make the spins want to line up with one another, called a ferromagnetic state. If J is negative, it will make the spins want to anti-align, leading to an anti-ferromagnetic state. The spins also want to line up in the same direction as the acting external field — they want to be positive if $B > 0$ and negative if $B < 0$.

The probability that the system is in one of those two states is given by

$$P = \frac{e^{-\beta H}}{Z} \tag{2.2}$$

with $\beta = 1/k_B T$, T the temperature and k_B is the Boltzmann constant.
The partition function of the Ising model is

$$Z = \sum_{\{s_i\}} e^{-\beta \mathtt{H}} \tag{2.3}$$

where $\{s_i\}$ means that we perform a sum for all spins and for all values of spins, which are $+1$ and -1. Hence, $\{s_i\}$ represents all possible micro-states of the Ising model.

The partition function can be interpreted as the normalization constant of the probabilities. It contains in itself all the statistical information about the system. The use of these probabilities is essential in implementing a Metropolis algorithm that can simulate the iterations between different spins and their respective switching behaviour.

The Ising model has two distinct phases, namely a paramagnetic phase in which its spins are disordered due to thermal fluctuations, and a ferromagnetic phase in which its spins start to preferentially align in one direction. These two phases are separated by a phase transition at some critical temperature $T = T_c$ below which the system becomes ferromagnetic. We can quantitatively distinguish these two phases by defining the magnetization

$$M \equiv \frac{1}{N} \sum_{i=1}^{N} \langle s_i \rangle \tag{2.4}$$

The magnetization serves as this system's order parameter, which means we can use it to describe how ordered the system is; for example, $M = 0$ corresponds to all of the spins being disordered (paramagnetic state) and $M \neq 0$ corresponds to the spins having a preferred direction (ferromagnetic state).

As we will later discuss, the Ising model can be seen as a very simple model for opinion dynamics, with agents being influenced by the state of the majority of their interacting partners.

2.1.1 Estimation of Expected Values

For a simulation of a system, the aim is to calculate the expected value of a quantity. For instance, the expected value of a quantity Q which follows a Maxwell-Boltzmann distribution is

$$\langle Q \rangle = \frac{\sum\limits_{\mu} Q_\mu e^{-\beta E_\mu}}{\sum\limits_{\mu} e^{-\beta E_\mu}}$$

where μ labels the state of the system and $\beta = \frac{1}{k_B T}$ and k_b is the Boltzmann constant.

When taking this method for calculating the expected value of a quantity, all states of the system must be considered and this can be realized for systems with a small number of states. However, when considering larger systems, in order to calculate the expected value of a quantity of system, a subset of these states must be used since all states cannot be considered. In this case, because of using a small portion of states, there will be an error in the calculated expected value. The Monte Carlo method allows us to get the expected value of a large system with low error.

Monte Carlo methods select states of the system from a probability distribution P_μ which we specify. With respect to the Monte Carlo method, the expected value of a quantity is given by

$$Q_M = \frac{\sum_{i=1}^{M} Q_{\mu_i} {P_{\mu_i}}^{-1} e^{-\beta E_{\mu_i}}}{\sum_{j=1}^{M} {P_{\mu_j}}^{-1} e^{-\beta E_{\mu_j}}} \tag{2.5}$$

where states of the system are $\{\mu_1, ..., \mu_M\}$ and Q_M is an *estimator of* Q. In the formulation, we see ${P_{\mu_i}}^{-1}$ because we chose state Q_{μ_i} by a probability P_{μ_i} from all of the states.

Now, for an accurate estimation of expected value, P_μ must be defined. For instance, P_μ may be chosen as equal for all states. Then,

$$Q_M = \frac{\sum_{i=1}^{M} Q_{\mu_i} e^{-\beta E_{\mu_i}}}{\sum_{j=1}^{M} e^{-\beta E_{\mu_j}}}$$

This is a poor choice for P_μ. If probabilities of some states are greater than others, in order to obtain a good estimation of the expected value, these states with high probabilities can be used. Hence, it is obvious that, if P_μ is chosen in the form of Maxwell-Boltzmann distribution, then a better approximation to the expected value will occur.

When the number of states used to calculate expected value of a quantity by Monte Carlo method is increased, then the estimator of Q converges to $\langle Q \rangle$.

$$\lim_{M \to \infty} Q_M = \langle Q \rangle$$

For a system with states which does not have equal probabilities, an appropriate P_μ must be defined in order to approximate the expected value of the system. We have specified that if we choose P_μ in the form of a Maxwell-Boltzmann distribution, then we get a better approximation of the expected value.

We choose P_μ to be of the form,

$$P_\mu = Z^{-1} e^{-\beta E \mu} \tag{2.6}$$

then

$$Q_M = \frac{1}{M} \sum_{i=1}^{M} Q_{\mu_i} \tag{2.7}$$

This sampling is called *importance sampling*. The partition function Z is a normalization constant for the probability P_μ. Since Z is known, then all macroscopic properties of the system can be calculated.

2.2 Markov Process and Selection Probabilities

In order to pick states so that each one appears with its correct Maxwell-Boltzmann probability, we implement multiple Markov Processes in our simulation.

The Markov Process is a mechanism found by the Russian Mathematician Andrey Markov. For a given state μ, the mechanism creates a new random state ν. The probability to move a state from μ to ν is the *transition probability*, $P(\mu \to \nu)$.

The transition probabilities must satisfy two conditions: they should be time independent and depend only on the initial and final states, μ, ν, respectively. Also, it is obvious that,

$$\sum_{\nu} P(\mu \to \nu) = 1 \tag{2.8}$$

that is, the sum of all probabilities of transitioning from a state μ to other states ν (and also the probability of staying in the same state) is equal to 1.

2.2.1 Markov Chain

When simulating a system with a Monte Carlo method, the Markov Process is used repeatedly, as a chain, creating what is known as a *Markov Chain*. The simulation cycles through the states, $\mu \to \nu \to \delta \to \lambda \to ....$

The Markov Process is chosen specially so that when it is run for long enough with any given initial state, it will eventually produce a succession of states which appear with probabilities given by the Maxwell-Boltzmann distribution.

In order to get a stable distribution at the end of a Markov chain, we should achieve the equilibrium condition. The equilibrium condition states that, transitions into a state and out of that state must be equal.

An equilibrium condition can be written as follows,

$$\sum_{\nu} P_{\mu} P(\mu \to \nu) = \sum_{\nu} P_{\nu} P(\nu \to \mu) \tag{2.9}$$

Therefore,

$$P_{\mu} = \sum_{\nu} P_{\nu} P(\nu \to \mu) \tag{2.10}$$

If the above equation is satisfied at any step of the Markov chain, then our probability distribution at the end of the chain will be P_{μ}. Now, we need a condition to achieve intended P_{μ} at equilibrium state. To achieve this intended P_{μ}, we should reach *simple equilibrium*, by avoiding *dynamic equilibrium*.

1. Simple Equilibrium:

   Let **P** be a matrix that consists of transition probabilities $P(\mu \to \nu)$, **P** is called a *Markov Matrix* or *Stochastic Matrix* and let $\omega_{\mu}(t)$ be the probability that the system is at a state μ at time t. Then,

   $$\omega_{\nu}(t+1) = \sum_{\mu} P(\mu \to \nu) \omega_{\mu}(t)$$

   Using matrix notation

   $$\underline{\omega}(t+1) = \mathbf{P}\underline{\omega}(t) \tag{2.11}$$

   $\underline{\omega}(t)$ is an array consisting of $\omega_{\mu}(t)$.

   If a simulation reaches simple equilibrium, then

   $$\underline{\omega}(\infty) = \mathbf{P}\underline{\omega}(\infty) \tag{2.12}$$

   In this case, $\underline{\omega}(\infty)$ equals a Maxwell-Boltzmann distribution, by choosing the appropriate probabilities.

2. Dynamic Equilibrium:

   If we reach dynamic equilibrium, the probability distribution ω rotates around different values. This rotation is called *limit cycle*. If we are in a limit cycle, we get

   $$\underline{\omega}(\infty) = \mathbf{P}^n \underline{\omega}(\infty) \tag{2.13}$$

   where n is the length of the limit cycle. To get rid of a dynamic equilibrium, we should satisfy the *detailed balance* condition.

3. Detailed Balance:

   By using detailed balance, we obtain a Maxwell-Boltzmann distribution after achieving the equilibrium condition. Detailed balance is given by the equation

   $$P_\mu P(\mu \to \nu) = P_\nu P(\nu \to \mu) \tag{2.14}$$

   The detailed balance condition clearly forbids dynamic equilibrium. For instance, we have a given probability for a state and this probability will increase at limit cycle. In order to increase this probability we must have more transitions to this state than out of it, on average; however, detailed balance prevents this from happening.

   If we look at the detailed balance condition, it also includes the equilibrium condition described above. If detailed balance is satisfied, then the equilibrium condition will be satisfied as well and we will avoid getting dynamic equilibrium and limit cycles.

   Once we remove limit cycles in this way, it is straightforward to show that the system will always tend to the probability distribution P_μ as $t \to \infty$. (citation p.38 Newman+Barkema). It means that $\omega_\mu(\infty) = P_\mu$.

   Let us now show that the system tends to P_μ as $t \to \infty$;

   It is known that, $\underline{\omega}(t+1) = \mathbf{P}\underline{\omega}(t)$. If t is chosen to be equal to zero, we will get $\underline{\omega}(1) = \mathbf{P}\underline{\omega}(0)$ and by making some iterations, then

   $$\underline{\omega}(t) = \mathbf{P}^t \underline{\omega}(0) \tag{2.15}$$

   Also, $\underline{\omega}(0)$ can be expressed as a linear combination of eigenvectors $\underline{v_i}$ of $\mathbf{P}$, [6]

   $$\underline{\omega}(0) = \sum_i a_i \underline{v_i} \tag{2.16}$$

   Then

   $$\underline{\omega}(t) = \mathbf{P}^t \sum_i a_i \underline{v_i}$$

   We know that $\mathbf{P}\underline{v_i} = \lambda_i \underline{v_i}$, where λ_i is the eigenvalue corresponding to the eigenvector $\underline{v_i}$. Then,

   $$\underline{\omega}(t) = \sum_i a_i {\lambda_i}^t \underline{v_i} \tag{2.17}$$

   As $t \to \infty$, ${\lambda_i}^t$ grows with t and the equation will be dominated by λ_0. This implies that $\omega(t)$ is proportional to $\underline{v_0}$ with its eigenvector corresponding to λ_0.

We know that $\sum_{\nu} P(\mu \to \nu) = 1$. Thus, the sum of all elements in each column of the Markov matrix equals to 1. On the other hand, since elements of the Markov matrix describe probabilities, they must be in the range $[0, 1]$.

Let's assume a 2x2 Markov Matrix **P**;

$$\begin{bmatrix} a & b \\ 1-a & 1-b \end{bmatrix}$$

its transpose is $\mathbf{P}^T$

$$\begin{bmatrix} a & 1-a \\ b & 1-b \end{bmatrix}$$

To find its eigenvectors and eigenvalues $\mathbf{P}^T v_i = \lambda_i \underline{v_i}$ then, $(\mathbf{P}^T - \lambda_i I)\underline{v_i} = 0 = \mathbf{K}\underline{v_i}$ then **K** is

$$\begin{bmatrix} a-\lambda & 1-a \\ b & 1-b-\lambda \end{bmatrix}$$

Hence, generally for the transpose of an nxn Markov Matrix **P**, the eigenvector $\underline{v_i}$ is

$$\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

and for this unit element eigenvector, there is an eigenvalue of 1. To see this, using the same 2x2 Markov Matrix, and $\mathbf{P}^T \underline{v_i} = \lambda_i \underline{v_i}$; $a - \lambda + 1 - a = 0$; $b + 1 - b - \lambda = 0$ then $\lambda = 1$.

The Markov matrix **P** and its transpose have the same determinant, then $\mathbf{P} - \lambda I$ and $\mathbf{P}^T - \lambda I$ also have the same determinant. Eigenvalues of **P** and $\mathbf{P}^T$ are the same (because the determinants are equal). Therefore, the Markov Matrix **P** has an eigenvalue of 1.

Now, let's assume v_i is an eigenvector corresponding to eigenvalue $|\lambda| > 1$ and $\mathbf{P}^n \underline{v_i} = |\lambda|^n \underline{v_i}$, so its length grows exponentially as $n \to \infty$. Hence, for large n, an element of $\mathbf{P}^n$ must be larger than one (to satisfy equality), which is impossible. Therefore, all eigenvalues of **P** are smaller or equal to 1.

Also, we have found

$$\underline{\omega}(t) = \sum_i a_i {\lambda_i}^t \underline{v_i}$$

as $t \to \infty$. Eigenvalues with absolute value smaller than 1 vanish and the right hand side of the equation is dominated by $\lambda_0 = 1$. For a large enough t, $\underline{\omega}(t)$ is proportional to $\underline{v_0}$ which is the eigenvector corresponding to λ_0 which has elements of 1. Hence,

$$\underline{\omega}(\infty) = a_0 \underline{v_0} \tag{2.18}$$

In equilibrium, we have that $P_\mu = \sum_\nu P_\nu P(\nu \to \mu)$, in vector notation, is equal to $\underline{p} = \mathbf{P}\underline{p}$. This $\underline{p}$ is a vector whose elements are P_μ. For an eigenvalue and its corresponding eigenvector of a matrix **A**, $\mathbf{A}\underline{v} = \lambda \underline{v}$. We can then write

$$\mathbf{P}\underline{p} = \lambda \underline{p}$$

, and $\underline{p}$ is a normalized eigenvector of **P** which has eigenvalue 1. It means that $\underline{v_0} = \underline{p}$

Therefore, $\underline{\omega}(\infty)$ is $\underline{p}$. Because of this, when t goes to infinity, $\underline{\omega}(t)$ goes to $\underline{p}$:

$$\lim_{t\to\infty} \underline{\omega}(t) = \underline{p} \tag{2.19}$$

By carefully choosing transition probabilities which satisfy the detailed balance condition, we can reach any probability distribution $\underline{p}$ in equilibrium.

Now, we want to satisfy the Maxwell-Boltzmann distribution in equilibrium condition. In order to do this we choose p_μ as the Maxwell-Boltzmann probabilities. By using detailed balance condition;

$$\frac{P(\mu \to \nu)}{P(\nu \to \mu)} = \frac{P_\nu}{P_\mu} = e^{-\beta(E_\nu - E_\mu)} \tag{2.20}$$

Also, to show that all states must be a part of the system, we should satisfy the *ergodicity* condition.

4. Ergodicity:

   With respect to the ergodicity condition, some of the transition probabilities may be zero. However, for any two states which have transition probabilities equal to zero, there must be at least one path of non-zero probability. Thus, all states in the system must be reachable by at least one path, it means that all states must be a part of the system.

   We must then satisfy three conditions:

   (a) $\frac{P(\mu\to\nu)}{P(\nu\to\mu)} = \frac{P_\nu}{P_\mu} = e^{-\beta(E_\nu - E_\mu)}$

   (b) $\sum_\nu P(\mu \to \nu) = 1$

   (c) Ergodicity

   If these three conditions are satisfied, then we get a Maxwell-Boltzmann distribution as an equilibrium distribution.

   There are a lot of transition probabilities to satisfy these conditions. There are also plenty of algorithms to implement these mechanics, such as the conventional *Metropolis algorithm*. However, a purpose-built algorithm can often give a much faster simulation than an equivalent standard algorithm, and the improvement in efficiency can easily make the difference between finding an answer to a problem and not finding one [6].)

   Such is the case with the modified Monte Carlo algorithm we used, where the probabilities of the opinion switch mechanism operate under the same conditions.

2.2.2 Acceptance Ratio and Selection Probabilities

We have two equations to satisfy, one is the ratio of transition probabilities and the other is the sum of transition probabilities to move a state to other states.

Let us consider a "stay-at-home" condition $\nu = \mu$. For this scenario, the ratio of transition probabilities equals 1, so the first condition (detailed balance) is satisfied, independently of $P(\mu \to \mu)$. We can then choose other transition probabilities more

easily. This flexibility comes from the $\sum_{\nu} P(\mu \to \nu) = 1$, and we can adjust any $P(\mu \to \nu)$. For an adjustment in $P(\mu \to \nu)$, we balance $P(\nu \to \mu)$ out to keep the ratio constant. So, by changing $P(\mu \to \mu)$, we can use any value of $P(\mu \to \nu)$.

Let's assume

$$P(\mu \to \nu) = g(\mu \to \nu)A(\mu \to \nu)$$

where $g(\mu \to \nu)$ is the *selection probability*, which is the probability of generating a new state and $A(\mu \to \nu)$ is the *Acceptance Ratio*. Acceptance ratios are about the "stay-at-home" conditions, for example if we choose $A(\mu \to \nu) = 0$ for all ν then $P(\mu \to \mu) = 1$, because if we don't accept any move to another state, then our "stay-at-home" probability is 1. So we are free to choose any value of $A(\mu \to \nu)$ in between 0 and 1.

If we look at the equation $\frac{P(\mu \to \nu)}{P(\nu \to \mu)} = \frac{P_\nu}{P_\mu} = e^{-\beta(E_\nu - E_\mu)}$, then we have a constraint in

$$\frac{P(\mu \to \nu)}{P(\nu \to \mu)} = \frac{g(\mu \to \nu)A(\mu \to \nu)}{g(\nu \to \mu)A(\nu \to \mu)}$$

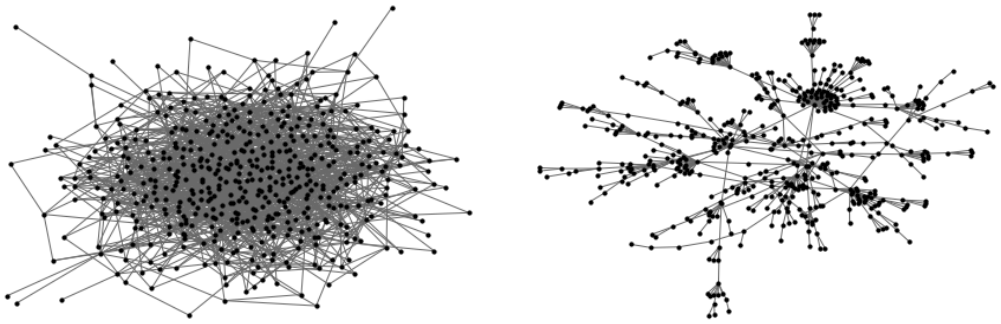
We can choose any value of $\frac{A(\mu \to \nu)}{A(\nu \to \mu)}$ and this gives us a freedom in the value of $g(\mu \to \nu)$ and $g(\nu \to \mu)$. Hence, we will create a MC algorithm that creates new states with probabilities $g(\mu \to \nu)$ and then we will accepts it with probabilities $A(\mu \to \nu)$.

However, there is also a requirement in the values of $A(\mu \to \nu)$. If we choose acceptance ratios to be low, then we cannot generate enough states. Thus, we want to choose $A(\mu \to \nu)$ as close as to unity. For chosen selection probabilities, we only have a constraint on the ratio $\frac{A(\mu \to \nu)}{A(\nu \to \mu)}$. Hence, we choose largest acceptance ratios as 1, that implies, for example, $A(\mu \to \nu) = 1$, and the other is fixed by the ratio of acceptance probabilities.

2.3 Graph Theory

The study of complex networks has traditionally been the territory of graph theory. While graph theory initially focused on regular graphs, since the 1950's large-scale networks with no apparent design principles were described as random graphs, proposed as the simplest and most straightforward realization of a complex network. Random graphs were first studied by the Hungarian mathematicians Paul Erdős and Alfréd Rényi [7]. According to the Erdős–Rényi model, we start with N nodes and connect every pair of nodes with probability p, creating a graph with approximately $pN(N-1)/2$ edges distributed randomly. This model has been the guiding model of the study of complex network decades after its introduction (see Fig. 2.1a). However, the growing interest in complex systems has led many to reconsider if real networks as diverse as the cell, the Internet, or social networks are fundamentally random. Our intuition and love for patterns would suggest that complex systems must display some underlying ordered structure which should be at somewhat evident in their topology. If the topology of these networks is indeed distinct of that of a random graph, we need to overview the means with which we quantitatively describe the underlying patterns.

In the past few years we have witnessed considerable advancements in this direction, prompted by multiple parallel developments. The large amount of data acquired by computerization led to the emergence of immense databases on the topology of a wide variety of real networks [9]. Increased computing power has allowed



(a) (b)

FIGURE 2.1: (a) A random Erdős–Rényi graph. (b) A Barabási–Albert graph showcasing its hierarchical structure and cluster formation. Made with [8].

the investigation into networks containing millions of nodes, opening doors to questions that could not be addressed before. There has also been a slow but noticeable breakdown of boundaries between disciplines. This has offered researchers access to diverse databases, allowing them to uncover the generic properties of complex networks. Finally, there is an increasingly voiced need to move beyond reductionist approaches and try to understand the behaviour of the system as a whole, searching for emergent properties. Motivated by these developments many quantities and measures have been proposed and investigated in depth in the past few years. The three main ones, that are essential for the understanding of this project, are *small-worldness*, *network clustering* and *degree of distribution*.

1. **Small Worldness:** In simple terms, the small world concept describes the fact that despite their often large size, most networks exhibit relatively short paths between any two nodes. The distance between two nodes is defined as the number of edges along the shortest path connecting them. The most popular manifestation of small worldness is the concept of "six degrees of separation", uncovered by the social psychologist Stanley Milgram [10], who concluded that there was a path of acquaintances with typical length of about six between most people in the United States [11]. The small world property appears as a characteristic of most complex networks: the actors in Hollywood are on average within three costars from each other, or the chemicals in a cell are separated typically by three reactions. While intriguing, this is not an indication of a particular organizing principle. The typical distance between any two nodes in a random graph scales as the logarithm of the number of nodes, as demonstrated by Erdős and Rényi. Thus random graphs are small worlds as well.

2. **Clustering:** A common property of social networks is the grouping of individuals into circles of friends or acquaintances in which every member knows every other member. This tendency to cluster in groups is quantified by the clustering coefficient [12]. Let us focus first on a selected node i in the network, having k_i edges which connect it to k_i other nodes. If the first neighbours of the original node were part of the same group, there would be $k_i(k_i-1)/2$ edges between them. The ratio between the number E_i of edges that actually exist between these k_i nodes and the total number $k_i(k_i-1)/2$ gives the value of the clustering coefficient of a given node i

$$C_i = \frac{2E_i}{k_i(k_i - 1)} \tag{2.21}$$

The clustering coefficient of the whole network is the average of all individual C_i's. In a random graph, since the edges are distributed randomly, the clustering coefficient is $C = p$. However, it was Watts and Strogatz who first pointed out that in most, if not all, real networks the clustering coefficient is typically much larger than it is in a random network of equal number of nodes and edges [13].

3. **Degree Distribution:** Not all nodes in a network have the same number of edges. The spread in the number of edges a node has, or node degree, is characterized by a distribution function $P(k)$, which gives the probability that a randomly selected node has exactly k edges. Since in a random graph the edges are placed randomly, the majority of nodes have approximately the same degree, close to the average degree $\langle k_i \rangle$ of the network. The degree distribution of a random graph is a Poisson distribution with a peak at $P(\langle k \rangle)$. Recent empirical results have shown that for most real networks the degree distribution significantly deviates from a Poisson distribution. These include the World-Wide Web [14], the Internet [15] or metabolic networks [16], the degree distribution has a power-law tail

$$P(k) \sim k^{-\gamma} \tag{2.22}$$

These are called scale-free networks [17]. While some networks do display an exponential tail, it is often the case that the functional form of $P(k)$ still deviates significantly from the Poisson distribution expected of a random graph.

These concepts have originated a boom in the field of network modelling. The three main focus points have been random graphs, small world graphs and scale-free graphs. Random graphs are variants of the Erdős–Rényi model widely used in many fields. They serve as a benchmark for many modelling and empirical studies, being the starting point of our attempted network modelling.

One of the most famous models for complex hierarchical random networks is the Barabási–Albert model [17]. Its construction starts from a small set of m fully interconnected nodes and new nodes are introduced one by one. Each new node selects m older nodes according to the preferential attachment rule, i.e., with probability proportional to their degree, and creates links with them. This procedure stops once the required network size N is reached. The network obtained has an average degree $\langle k_i \rangle = 2m$, a small clustering coefficient (of order $1/N$) and follows a power law degree distribution $P(k) \sim k^{-\gamma}$, with $\gamma = 3$. They also conclude that networks that have degree distributions with $\gamma \leq 3$ are scale-free (see Fig. 2.1b).

However, despite being the entry point for studying graphs, random graphs are not the best description for the topology of a social network. The class of small world models followed the discovery of clustering. These models interpolate between the highly clustered regular lattices and random graphs. Finally, the discovery of the power-law degree distribution has led to the construction of various scale-free models that, by focusing on the network dynamics, aim to explain the origin of the power-law tails and other non-Poisson degree distributions seen in real systems. These models obtain the network topology by executing the network dynamics.

Chapter 3

Opinion Dynamics

Opinion dynamics is the study of how groups of individuals adjust their beliefs and opinions as a result of interactions with one another, and exposure to additional information, media, or propaganda. The opinions held by individuals are expressed as variables - either real numbers within set bounds for continuous models, or a selection from a finite set in the case of discrete models. Relationships between individuals are modelled as edges on a graph, and rules to govern their interactions are introduced to represent real life. In each time step of the simulation, a group of agents interact with one another, and as a result shift their opinions towards or apart from one another.

Physicists have employed this type of approach to modelling social systems based on molecular dynamics [18, 19] and Metropolis and Monte Carlo [20] simulations. The main objective of these simulations is to address the problem of emergence from the lower (micro) level of the social system to the higher (macro) level.

The dynamics of opinion formation among individuals is a complex subject, because the individuals themselves and their motivations are. Modelling of social systems is a non-trivial matter and statistical physicists working on opinion dynamics aim at defining the opinion states of a population, and the elementary processes that lead to transitions between such states. The main question is whether this is possible and whether this approach can accurately describe human populations or if its merely a mathematical curiosity. Any such mathematical model must therefore consider opinion has a variable, or a set of variables, that is simple to manipulate. This may appear too reductive, when we consider the complexity of a person and of each individual position one can hold. However, multiple binary examples exists in day-to-day life that where people are given a limited set of choices on specific issues, which often are as few as two: right/left, buy/sell, PC/Mac, etc. Therefore, if these can be represented by a simple ± 1 representation, the challenge lies solemnly on finding an adequate set of mathematical rules that describe the mechanisms responsible for their evolution.

The first physicist to attempt to model opinion dynamics formation was based on a probabilistic framework [21], with the Ising model making its first appearance in the field a decade later [22, 23]. In such models, the spin-spin coupling represents the pairwise interaction between individuals, while the magnetic field represents the enforced cultural majority or media propaganda. Individual fields are also added, in an attempt to determine personal preferences towards either orientation. Depending on the strength of these individual fields, the system may reach a total consensus toward one of the two possible opinions (ferromagnetic), or a state where both opinions coexist with a certain ratio (paramagnetic).

The development of the field of opinion dynamics has been largely uncoordinated and based on individual attempts, where social mechanisms considered reasonable by the authors turned into mathematical rules, without a general shared

framework and often with no reference to real sociological studies or backing by big data. Nevertheless, attempts at unifying these frameworks have been done in recent years and interdisciplinary cooperation between physicists working on opinion dynamics and sociologists is (slowly) becoming more prevalent [24, 25, 26].

3.1 Opinion Formation in a Social Network

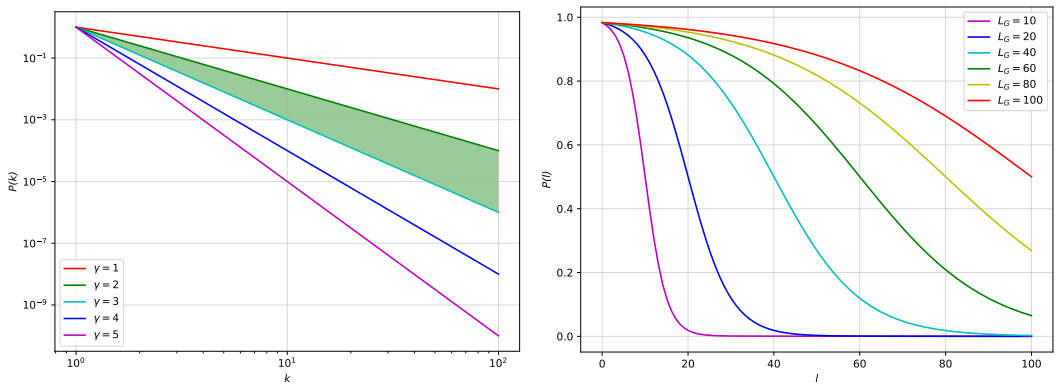
Over the past decades, physicists have actively worked on the field of complex networks and social dynamics, developing a wide range of models. An exhaustive description of these models is beyond the scope of this thesis, but we mention here those that have received the most attention in the physics literature over the past years and that we've considered using: the voter model [27, 28], the q-voter model [29], majority rule model [30], Sznajd model [31], amongst others [32]. Ultimately, most of these models did not allow for a convenient mechanism with which to externally shape opinion formation of the network, relying solemnly on the internal mechanisms acting between individuals. The topology of the networks used is also important, and one cannot rely on randomly generated graphs to provide an accurate representation of a real society.

We make use of an Ising-like opinion formation model that allows for external influences analogous to a magnetic field to affect the network's behaviour [33]. This particlar model also makes use of a network topology that is hierarchical and scale free, in an attempt to be an accurate aproximation of real social networks. Here we will present a description of this model, its equations and our implementation through a Python algorithm, which makes use of auxiliary modules to simulate graphs [8]. We also exhaustively investigate the properties of networks generated with our algorithm and its efficiency at doing so.

3.2 Network Structure

We describe the population as a two-dimensional lattice made up of $N = L \times L$ individuals. The graph structure is evenly spaced and arranged in such a way that each node is identified by the indices (i, j), which correspond to their fixed locations in the lattice. Each individual can have one of two states $S_{ij} = \pm 1$ assigned to them which correspond to their opinion regarding a certain topic or question. For our purposes, the opinion S_{ij} corresponds to their choice of political party. Individuals are randomly assigned one of two states in such a way that the average opinion at the start is $\langle S \rangle = 0$.

To accurately describe the way human populations interact we have to take into account the location of individuals within the lattice and their respective social connections with each other. These interpersonal interactions (social connections, friendships and random contacts) have a hierarchical structure. The distinction between first and second level connections is subsequently explained. The network of social connections is scale-free, meaning that the distribution of connectivity of individuals has the form $P(k) \sim k^{-\gamma}$, with $k \in (k_{min}, k_{max})$. Here we make a distinction between the connectivity of a node k_{ij} and its maximum allowed connectivity K_{ij}. The values of the connectivity distribution are assigned to each node and they correspond to the maximum number of connections that each individual is allowed to form, K_{ij}. Initially, the grid has no connections between individuals and therefore the actual value of connectivity for each node is $k_{ij} = 0$. Connections are then formed over multiple iterations over all nodes of the network until $k_{ij} = K_{ij}$ (or until no new connections



(a) (b)

FIGURE 3.1: (a) Power laws for different exponents γ. The region shaded in green corresponds to the range usually seen in both natural and artificial networks: $2 < \gamma < 3$. (b) Probability to form first level connections in function of the distance l between nodes for a lattice with $L = 100$ and multiple values of L_G. The second term in Eq. (3.1) causes $P(l)$ to reach zero slowly enough and parameters a and b ensure most connections are formed within the local group.

are possible, that is, the network is saturated). To generate the network of connections between individuals, we iterate over all nodes S_{ij} and pair them to a node S_{nm}, where $n = i \pm l_1$; $m = \pm l_2$ and $l = \sqrt{(i-n)^2 + (j-m)^2}$ is the distance between the nodes (l_1, l_2 are two independent random variables and the sign is generated with probability 0.5).

First level connections are created between individuals S_{ij} and S_{nm} based on a distance probability rule given by equation:

$$P(l) \sim \frac{1}{1 + \exp[(l-a)/b]} + 0.001\frac{L-1}{L} \tag{3.1}$$

The population is divided into local groups of $N_G = L_G \times L_G$ individuals, where the parameters $a = L_G$ and $b = L_G/4$ of Eq. (3.1) ensure that most of the first level connections created are in the same local group. A simple example of the structure of the network from the point of view of a certain individual is depicted in Fig. 3.2a. Whenever a first level connection is formed between nodes S_{ij} and S_{nm}, connections between S_{nm} and each neighbour of S_{ij} are formed with probability p_c (Fig. 3.2b). Similarly, attempts to form connections between S_{ij} and neighbours of individual S_{nm} are done, also with probability p_c. Connections formed this way are second level connections and through this procedure they influence the clustering coefficient of the nodes and, consequentially, the network. The clustering coefficient of the network is defined as the average

$$C = \left\langle \frac{2E_{ij}}{k_{ij}(k_{ij}-1)} \right\rangle \tag{3.2}$$

where E_{ij} is the number of connections between neighbours of the node (i, j).

Connections between two individuals can only be created once and new connections are only formed if the number of connections of an individual k_{ij} is lower than its maximum allowed number of connections K_{ij} (that is: $k_{ij} < K_{ij}$). This ensures that we obtain the desired distribution of connectivity for the network. We also point out that individuals with larger k cannot form connections with individuals with smaller

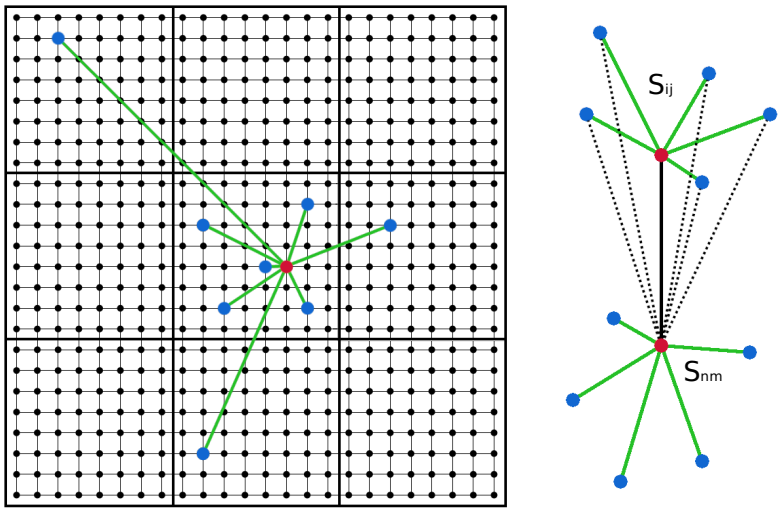


FIGURE 3.2: (a) Structure of a simple network with $L = 24$ and $L_G = 8$ from the point of view of the individual $S_{13,14}$ (in red), which has $k_{13,14} = 8$ connections (green lines) with its neighbours (in blue). The network has nine local groups and five of the connections are within the local group. (b) Distinction between first and second level connections. When a first level connection (black line) is formed between individuals S_{ij} and S_{nm}, the individual S_{nm} attempts to form second order connections (dotted lines) with the neighbours of S_{ij}, with probability p_c. Subsequently, connections between the neighbours of S_{nm} and S_{ij} are created (not depicted).

k. This means individuals with high connectivity act as hubs and are forced to create connections between each other. For large enough p_c we have highly interconnected groups.

3.3 Network Properties

We now look at some of the statistical properties of the network and how its multiple parameters influence it. We can see in Fig. 3.4a that the distribution of connectivity $P(k)$ depends only on the parameter γ, which is to be expected since our algorithm for creating network connections will add edges to a node but not after its number of connections k_{ij} reaches the maximum allowed number of connections K_{ij}, drawn at the start from the power law distribution $\sim k^{-\gamma}$. In Fig. 3.4b we can see that the distribution of connectivity is unaffected by the size of the local group L_G. It's worth pointing out that for increasingly higher values of γ the distribution will have higher scattering for larger values of connectivity k. This is can be a bit problematic for our algorithm, as we will later discuss. The relation between the clustering coefficient of a node and its number of connections is shown in Fig. 3.5 [we use a grid of $L = 100$ because of computational limitations - I'd like to repeat this for $L = 1000$ to get better statistics and less scattering]. Here we can see how the parameter p_c influences the clustering coefficient of the nodes

[for a larger network $L = 1000$ multiple values of L_G should also be plotted to see how little the size of the local group influences clustering. For now, I am skipping this since generating connections for large networks is very slow. I also want to fit the

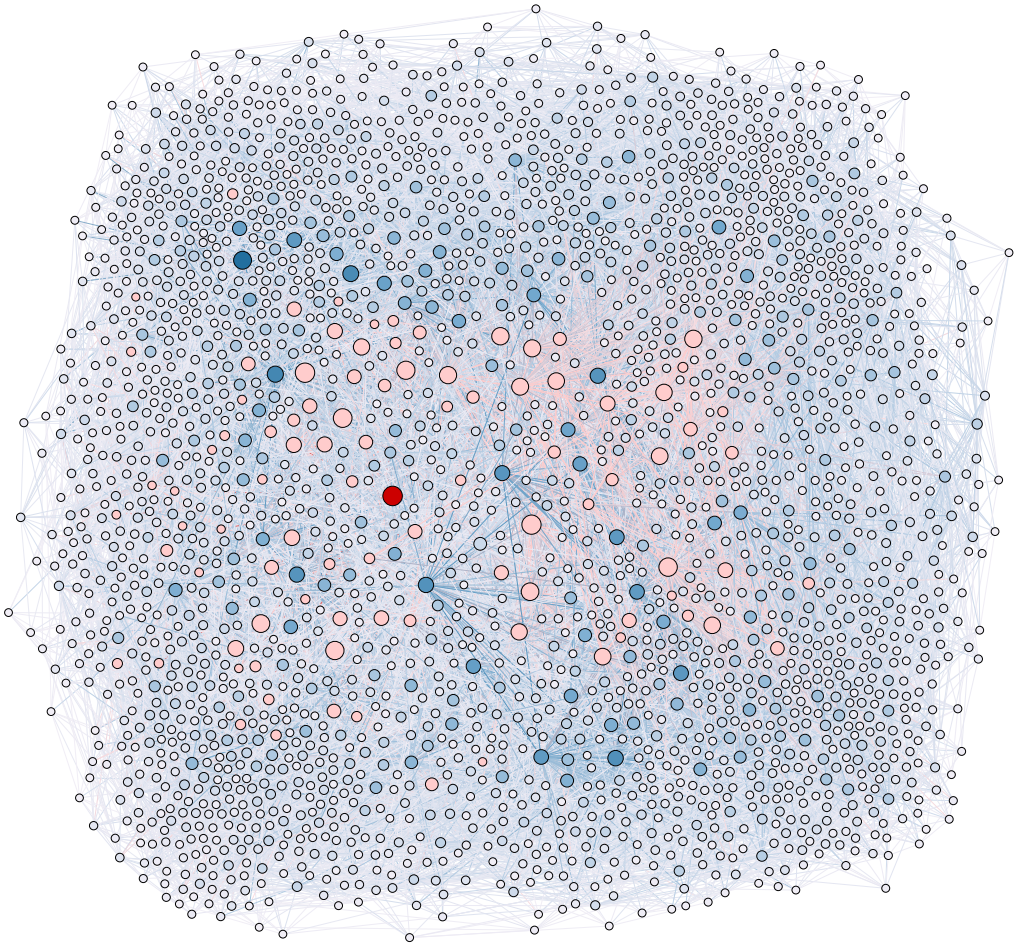


FIGURE 3.3: Visualization of a small network ($L = 50$) done with *Gephi* [34]. Upon the execution of our network generator algorithm, the grid layout can be rearranged so that the scale-free structure of the network is more evident. The size of every node corresponds to the relative degree of a node (its number of connections/neighbours k_{ij}). The node with highest degree ($k_{8,22} = 97$) is highlighted in red, along with all of its neighbours. Nodes with low degree are pushed to the periphery of the graph while the nodes with high degree form clustered hubs concentrated at the centre. The structure of the network is such that hubs are formed in a hierarchical manner, with low degree nodes linking to increasingly higher degree nodes in a tree-like way. The average degree of the network is $\langle k \rangle = 17.522$, with corresponding power law connectivity distribution $\sim k^{-\gamma}$. The network clustering coefficient is $C = 0.202$. Fixed parameters are $\gamma = 3$; $L_G = 20$; $k_{min} = 10$; $k_{max} = 100$; and $p_c = 0.5$.

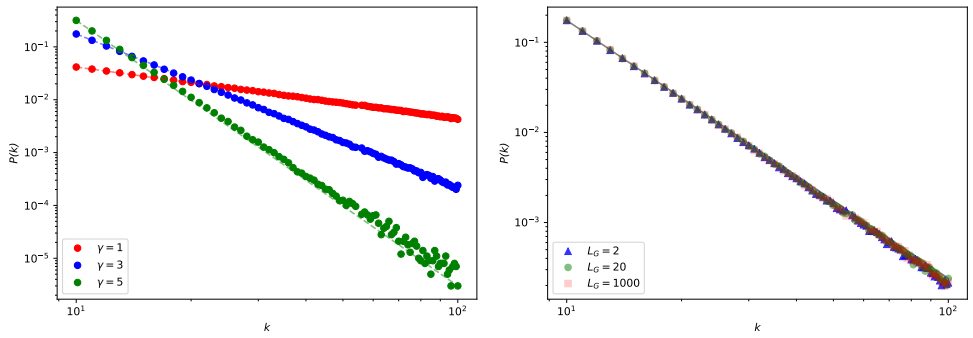


FIGURE 3.4: (a) Distribution of connectivity $P(k)$ for $L_G = 20$. Power law fits for the distributions presented are, respectively: $\gamma_1 \approx 0.98957$; $\gamma_3 \approx 2.86341$; and $\gamma_5 \approx 5.02414$. (b) Distribution of connectivity $P(k)$ for fixed $\gamma = 3$ and different sizes of the local group. We can easily see that the size of the local group has no effect on the distribution. Fixed parameters on both plots are $L = 1000$; $k_{min} = 10$; $k_{max} = 100$; and $p_c = 0.5$

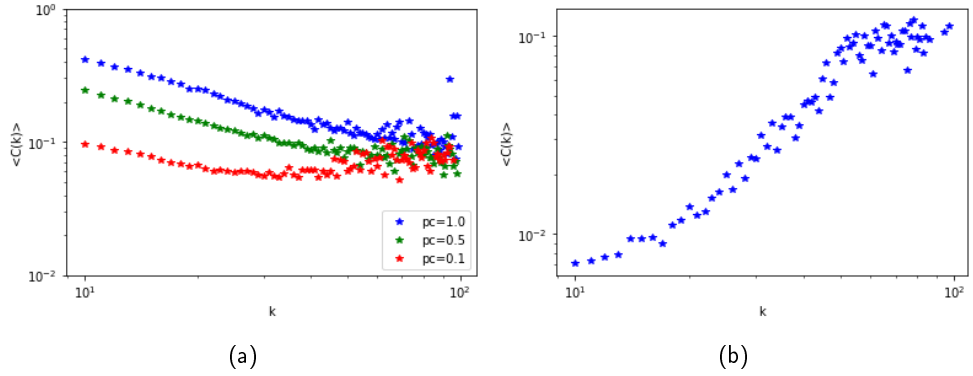


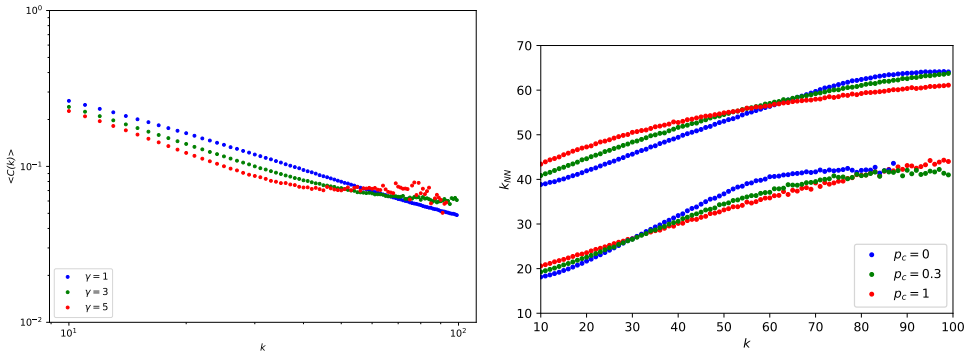
FIGURE 3.5: Average clustering coefficient of a node in relation to its number of connections k. Fixed parameters on both plots are $L = 100$; $k_{min} = 10$; $k_{max} = 100$; and $L_G = 20$ [re-do for a larger network]

results of clustering coefficient to a power-law $C(k) \sim k^{-\alpha}$ to check if it matches with the conclusions of the paper, since such power-law is observed in real networks]

Fig. 3.6a shows that the exponent α is slightly influenced by the parameter γ. Scattering for higher values of connectivity is high for two main reasons. First of, because there are only a few nodes with very high connectivity. This effect can be lessened in larger networks, since the power law distribution for such systems will allocate a larger number of high connectivity nodes. Secondly, larger networks also allow the generator algorithm more opportunities to create primary and secondary connections for these high connectivity nodes. The effect can be seen for the case where $\gamma = 5$, where there's noticeable scattering due to the low probability of high connectivity nodes occurring.

3.4 Opinion-Switch Mechanism

To model the opinion dynamics of the network we make use of an Ising-like approach. Each individual is influenced by the local field h_{ij}, which depends on the interactions with its neighbours, as follows:



(a) (b)

FIGURE 3.6: (a) Average clustering coefficient of a node in function of its number of connections k for fixed $p_c = 0.5$ (b) Average connectivity k_{NN} of nearest neighbours of a node in function of the number of connections k. The upper values correspond to $\gamma = 1$ and values underneath (with large scattering) are for $\gamma = 3$. Fixed parameters on both plots are $L = 1000$; $k_{min} = 10$; $k_{max} = 100$; and $L_G = 20$.

$$h_{ij}(t+1) = \frac{1}{k_{ij}} \left(\sum_{n=1}^{k_{ij}} A_{l(n)m(n)} S_{l(n)m(n)}(t) + \frac{1}{N_G} \sum_{n=1}^{N_G} A_{l(n)m(n)} S_{l(n)m(n)}(t) \right) + I_E \quad (3.3)$$

Opinion formation of an individual (i, j) is then controlled by the interpersonal interactions with its k_{ij} neighbours (the first term in Eq. (3.3)) and the influence of their local group (the second term in Eq. (3.3)). The first term is the most effective one, while the influence of the local group has a weaker role. The added parameter I_E is the external stimulation, which will be discussed in detail later. The parameter A_{ij} is the authority of the individual (i, j) which is a property of each node, similar to S_{ij}. The values of the authority are generated from a truncated Gaussian distribution in the range $A_{ij} \in (0, 1)$ with mean value $\mu = 0$ and variance σ. This property of the node corresponds to an individual's influence over other individuals regarding opinion formation. Higher authority A_{ij} means that the opinion of individual (i, j) has higher importance in the social community. We can then describe the probability p_{ij} that an individual changes their opinion (the value of S_{ij}) based on the authority and local field:

$$p_{ij} = \begin{cases} (1 - A_{ij})(1 - \exp[h_{ij} S_{ij} / T]); & h_{ij} S_{ij} \leq 0 \\ (1 - A_{ij}) \exp[-h_{ij} S_{ij} / T]; & h_{ij} S_{ij} > 0 \end{cases} \quad (3.4)$$

where T is the statistical physics parameter for temperature, here interpreted as some sort of measure of "social unrest". For higher values of A_{ij}, the less likely it is that the individual will change their opinion. With greater h_{ij}, the greater the probability that an individual will have the opinion that conforms with the local field. On the other hand, for increasing temperature T there is an increase in the probability that the individual will oppose the opinion of the local field h_{ij}. For the case when $T = 0$, Eq. (3.4) simplifies to:

$$p_{ij} = \begin{cases} (1 - A_{ij}); & h_{ij} S_{ij} < 0 \\ 0; & h_{ij} S_{ij} \geq 0 \end{cases} \quad (3.5)$$

Therefore, when the opinion of an individual is in agreement with the local field ($h_{ij}S_{ij} > 0$) the probability of switching opinion equals to zero.

3.5 Simulation Results

We run all simulations from initial conditions where the network is settled in a paramagnetic phase ($\langle S \rangle \equiv N^{-1}\sum_{i,j}^{L} S_{ij} = 0$) and then run a Monte Carlo method that swipes through the entire network and gives random individuals a go at attempting to convince their neighbours to conform to their opinion S_{ij}. Effectively, this implies that every one of our Monte Carlo steps corresponds to N^2 attempts at opinion switching. This can be done in two ways: One can either swipe the entire network and let every single individual enforce their opinion S_{ij} on their neighbours, therefore consulting all the nodes at each Monte Carlo step; or alternatively, one can randomly select nodes as is done in conventional atomic simulations. There is, however, a concern that a certain bias could emerge if a lattice swipe is performed, even though both methods were tested and shown to have similar results when averaged over multiple independent simulations. Regardless, for the simulations shown here, the node selection is always random. A visual representation of the matrices associated with the main network properties is shown in Fig. 3.7.

3.5.1 Acceptance Rates

We start by analysing the acceptance rates of the algorithm used for our network generation. This is relevant for the purposes of confirming that the algorithm is not only creating connections as expected, but also for choosing the number of iterations required to obtain a final network that matches the theoretical power law distribution $P(k) \sim k^{-\gamma}$. Since allocation of connections is based on Eq. 3.1, the distance between nodes is a crucial factor. As nodes create connections between them, their maximum allowed number of connections is reached - this is not necessarily true for nodes with large K_{ij}, and ends up being critical to obtaining a network that matches $P(k)$ sufficiently well. In Fig. 3.8 we show the acceptance rate for four different instances of p_c over the course of 10000 generator iterations. For the case where $p_c = 0$, only primary connections are generated (Fig. 3.8a). Networks built under these parameters have very low connectivity and take a considerable amount of time to match the power law distribution $P(k)$. Most notably, the generator algorithm slows down considerably once nodes that are spatially close to each other have their maximum number of connections allocated. Since under these conditions the distance between nodes is the only factor influencing edge creation, it becomes increasingly unlikely for the network to reach a saturation that is within a satisfactory agreement with $P(k)$. This turns out to be a significant limitation of the algorithm. The subsequent cases are for a non-zero value of p_c and therefore we allow for secondary connections to be formed. Most connections are formed within the first 1000 iterations and the most clear observation we can make is that for low p_c (Fig. 3.8b) the number of secondary connections formed never surpasses the number of primary connections. For higher p_c (Fig. 3.8c and 3.8d) we see that secondary connections overtake primary connections quite quickly, and this is very important for the network to reach the required power law distribution $P(k)$ within a reasonable amount of computational time. For illustrative purposes, Fig. 3.9 and 3.10 shows how most connections are formed within the first 250 iterations of the algorithm.

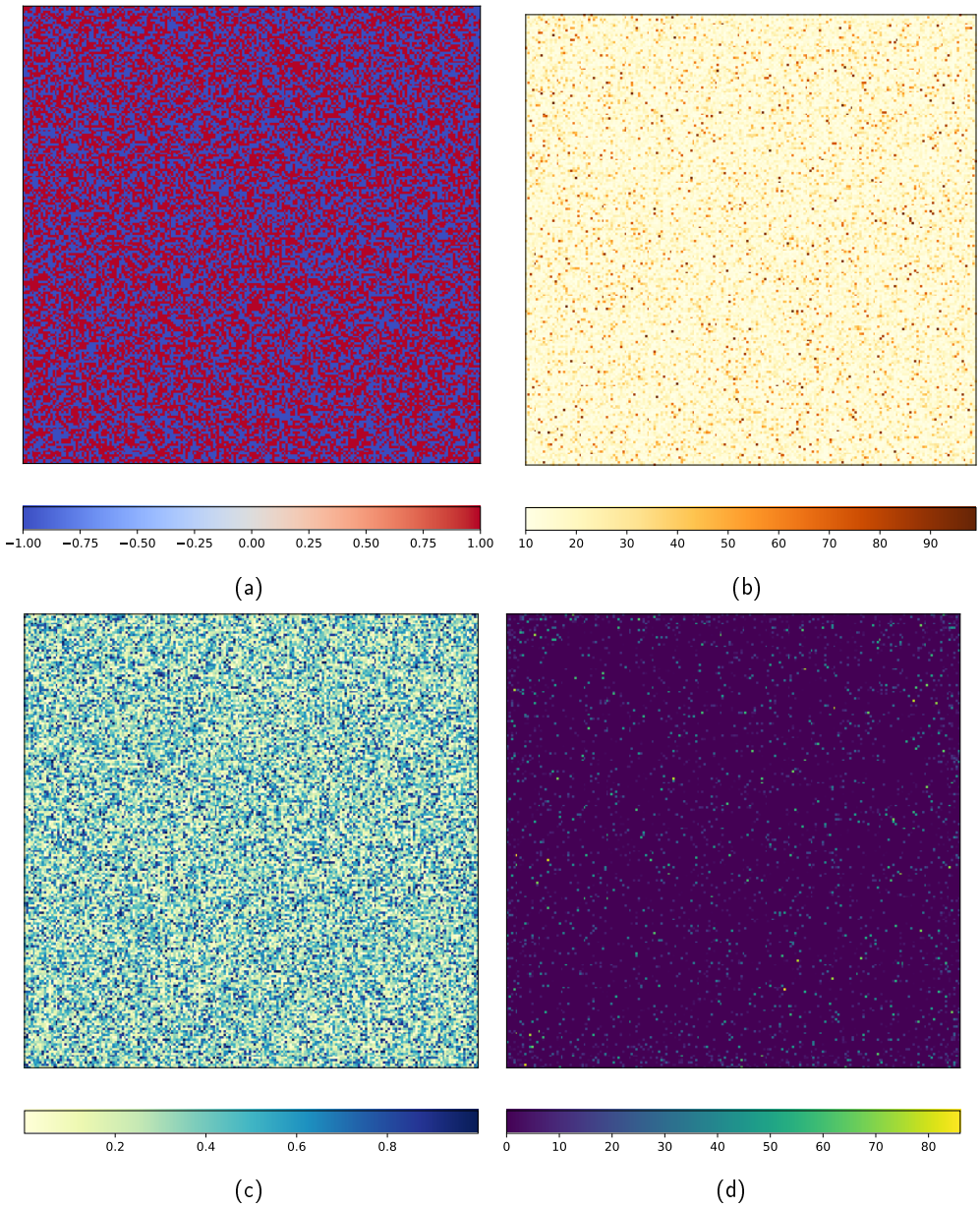


FIGURE 3.7: Visual representations of the network properties for $L = 200$; $\gamma = 3$; $k_{min} = 10$; $k_{max} = 100$; $p_c = 0.5$; and $L_G = 20$. Each node of the grid is represented as a coloured pixel. (a) Opinion S_{ij}, with $S_{ij} = 1$ in red and $S_{ij} = -1$ in blue. (b) Maximum number of connections K_{ij} drawn from the distribution $\sim k^{-\gamma}$, where the values are integers in the range (k_{min}, k_{max}). (c) Authority A_{ij}, where values are drawn from a truncated Gaussian distribution in the range $(0, 1)$. (d) Network saturation (here simply the difference $K_{ij} - k_{ij}$) after 1000 iterations of the connection generator algorithm.

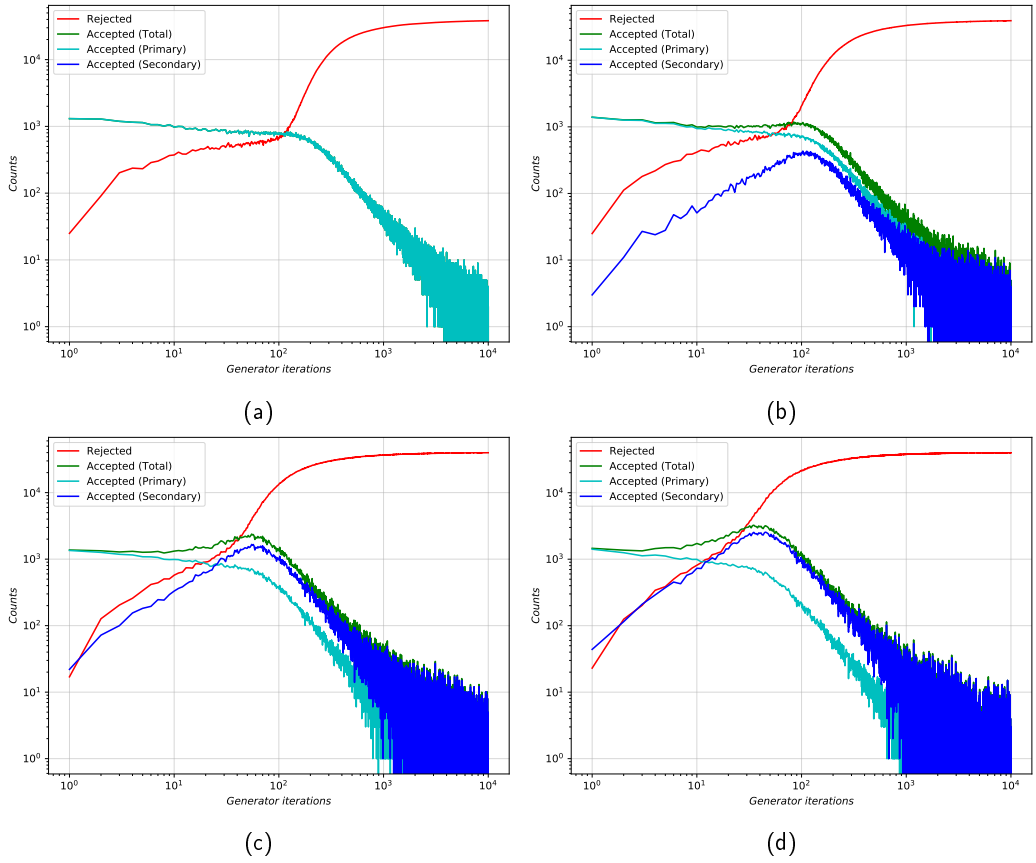


FIGURE 3.8: Acceptance rates for the network generator algorithm for different values of p_c, subsequently affecting the final network clustering. Generator iterations are set at 10000, with each iteration cycling over all nodes ($L^2 = 40000$) until it either succeeds in forming an edge or fails. Fixed parameters are $L = 200$; $\gamma = 3$; $k_{min} = 10$; $k_{max} = 100$; and $L_G = 20$. (a) $p_c = 0$ and $C \sim 0.0081$ - note that here, only primary connections are generated. (b) $p_c = 0.1$ where $C \sim 0.0771$. (c) $p_c = 0.5$ where $C \sim 0.1803$. (d) $p_c = 1.0$ where $C \sim 0.3110$.

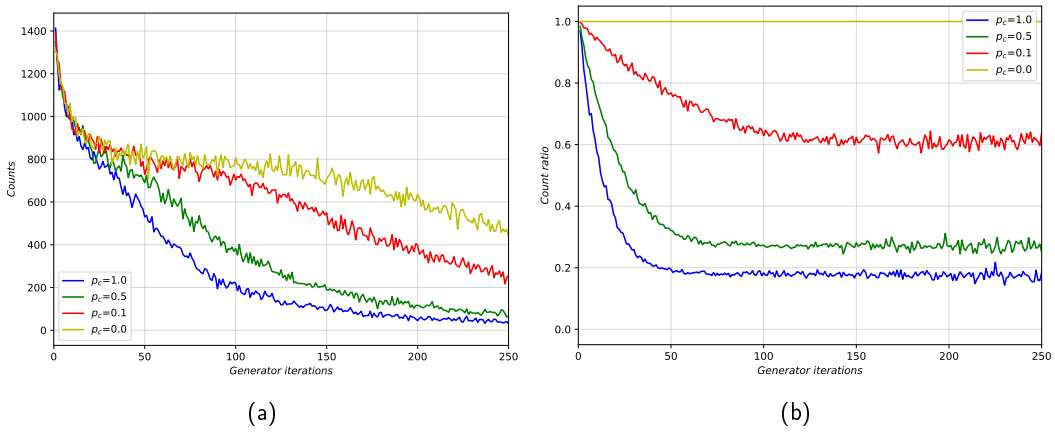
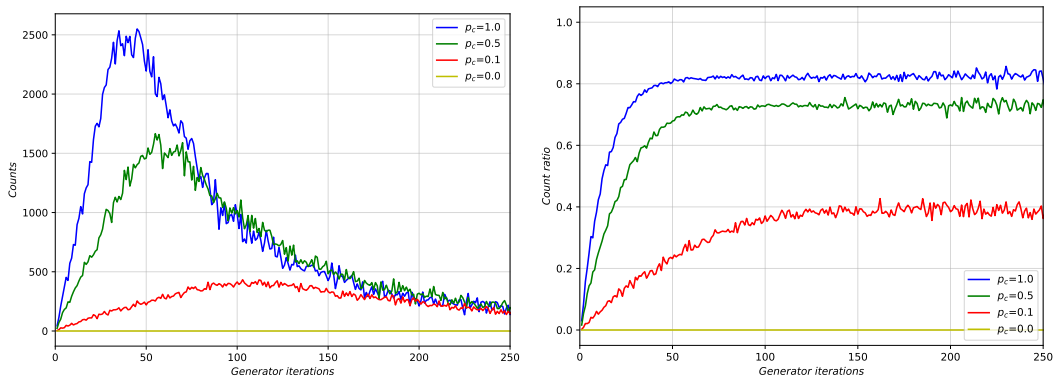
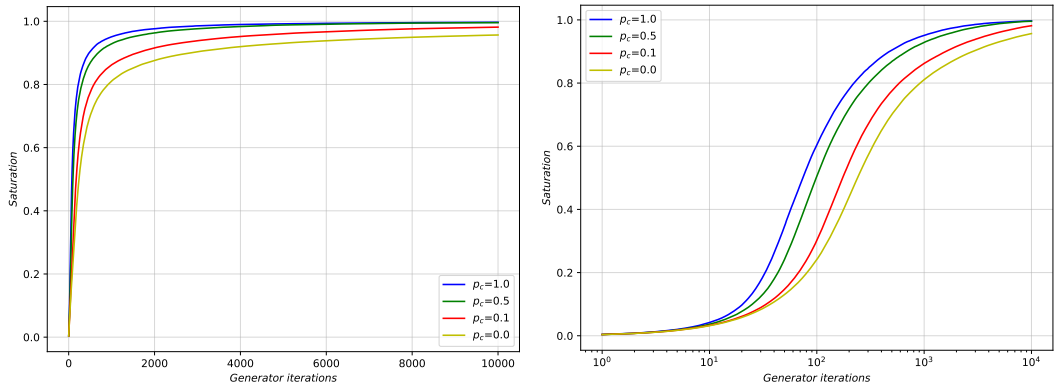


FIGURE 3.9: Primary connections generated over the first 250 iterations for different values of p_c. (a) Number of edges generated at each iteration. (b) Ratio of secondary connections and total number of generated connections for each iteration. Fixed parameters are $L = 200$; $\gamma = 3$; $k_{min} = 10$; $k_{max} = 100$; and $L_G = 20$.



(a) (b)

FIGURE 3.10: Secondary connections generated over the first 250 iterations for different values of p_c. (a) Number of edges generated at each iteration. (b) Ratio of secondary connections and total number of generated connections for each iteration. Fixed parameters are $L = 200$; $\gamma = 3$; $k_{min} = 10$; $k_{max} = 100$; and $L_G = 20$.



(a) (b)

FIGURE 3.11: Network saturation over 10000 iterations for different values of p_c. Fixed parameters are $L = 200$; $\gamma = 3$; $k_{min} = 10$; $k_{max} = 100$; and $L_G = 20$.

3.5.2 Network Saturation

To ensure that a network matches the power law distribution $P(k)$ with a reasonable level of satisfaction, we devise a way to check how much of the theoretically available space for connections has been filled up by our algorithm. To do this, we measure the saturation of the network by simply comparing the maximum number of connections allocated to each node by the distribution $P(k)$ (denoted K_{ij}) with the actual number of connections of each node as built by the algorithm (denoted k_{ij}). This is simply k_{ij}/K_{ij} and we want this to be as close as possible to 1. We set a minimum threshold for saturation to be of 0.998 for most of our shown networks. This is necessary, as otherwise the nodes with higher connectivity will not match the power law distribution. Networks with low p_c and consequently low clustering coefficient C are notoriously difficult to be made to meet this criteria. The saturation of different networks over the course of the first 10000 iterations is shown in Fig. 3.11 in both linear and logarithmic scales. One can see that only networks with $p_c > 0.5$ are sufficiently close to 1 and our required threshold. Since the saturation is asymptotic, we can say that networks with $p_c < 0.5$ will never meet this requirement and therefore their "quality" is questionable at best, which is a limitation of our algorithm. Despite

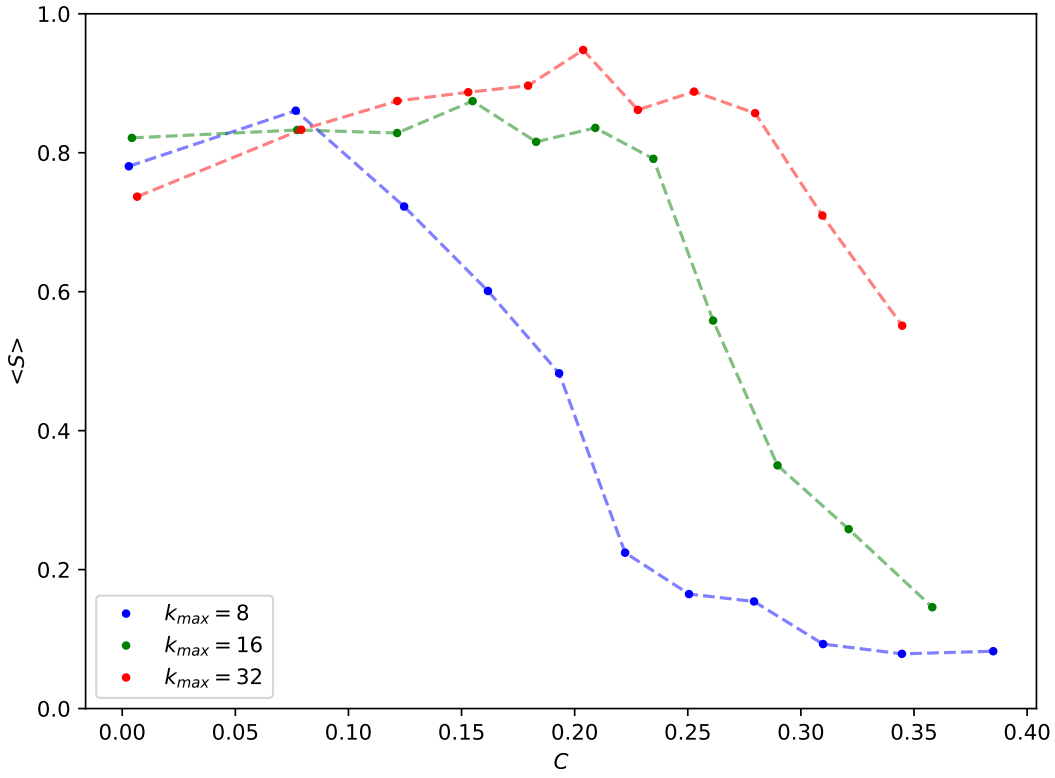


FIGURE 3.12: Relation between $\langle S \rangle$ and the clustering coefficient C for different values of k_{max} and $T = 0$. Values of other parameters are $L = 100$; $L_G = 20$; $k_{min} = 8$; $\sigma = 0.3$; and $I_E = 0$. Results are averaged over 100 independent simulations. The number of time steps of each simulation is 1000.

that, since real networks have $p_c \sim 0.5$, this will not be a problem.

3.5.3 Network Clustering

We now focus our attention on the effect of the clustering coefficient C on the value of $\langle S \rangle$. For a small value of C, in most cases, the system settles into a ferromagnetic state ($\langle S \rangle = 1$). However, for increasing values of clustering coefficient, $\langle S \rangle$ decreases, as seen in Fig. 3.12. One would still expect that a higher number of simulations would reach a ferromagnetic phase for low clustering coefficient, but this is not shown to be the case. Two main points contribute to this disparity: the topological quality of the network generated (which for low clustering coefficient C means a low value of p_c), and the authority matrix A_{ij}. As we have previously discussed, there are limitations when trying to generate a network that matches the theoretical power law distribution for low p_c. Take $p_c = 0$; the only type of connections generated are first order connections. This means that all attempts at creating connections are dependant on the distance probability and, when employing all the conditions for generating said connections, filling all available connections for nodes with high K_{ij} is very unlikely to happen within a reasonable simulation time. Network quality is then hindered. The quality of the network is measured by the network saturation. Another factor is the authority matrix A_{ij}. As the opinion switch mechanism is inherently random, it is perfectly possible for an individual with high authority to quickly subjugate the entire network, or at least the vast majority. This authority

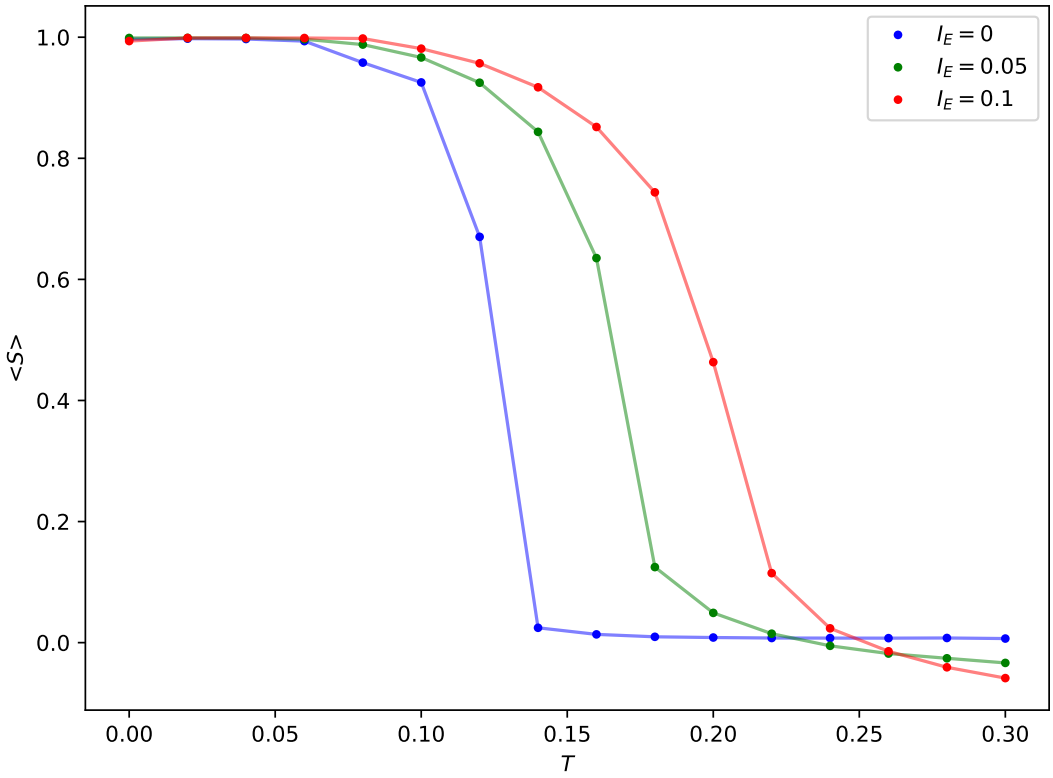


FIGURE 3.13: Relation between $\langle S \rangle$ and the network temperature T for different values of I_E. Values of other parameters are $L = 100$; $L_G = 20$; $p_c = 0.5$; $k_{min} = 8$; $k_{max} = 24$; and $\sigma = 0.3$. Results are averaged over 100 independent simulations. The number of time steps of each simulation is 1000.

bias seems to be amplified under these extreme cases, when connectivity is low and the system is then prone to settle into local minima.

For large C the population is divided into clusters (domains) of highly connected individuals with the same state. The individuals from different domains weakly interact among themselves.

3.5.4 External Stimulation

The main motivation for working with this specific model of opinion formation was because it offered us a simple way to externally influence the opinion of individuals within the network. This is done with the external influence parameter I_E, which acts on the local field h_{ij} analogously to an external magnetic field on an Ising magnet (see Eq. (3.3)). We are therefore interested in exploring how the network reacts to it and how it affects the average opinion formation. Fig. 3.13 shows how the average opinion $\langle S \rangle$ changes under a constant I_E for different network temperatures T when starting from a paramagnetic phase. We can see that the for low network temperatures the system always settles in a ferromagnetic phase. As the network temperature increases, we can see a phase transition towards a paramagnetic phase, which occurs at a critical temperature T_c. The system can settle into $\langle S \rangle = \pm 1$, but for the sake of presenting our results we only consider $\langle S \rangle > 0$ states while $T < T_c$ since we take $I_E > 0$ (results for $I_E < 0$ are the same). For $T > T_c$, the system enters a paramagnetic phase and here we consider all values of $\langle S \rangle$ into our data. This is done so we can better understand the impact of external influence I_E on forming

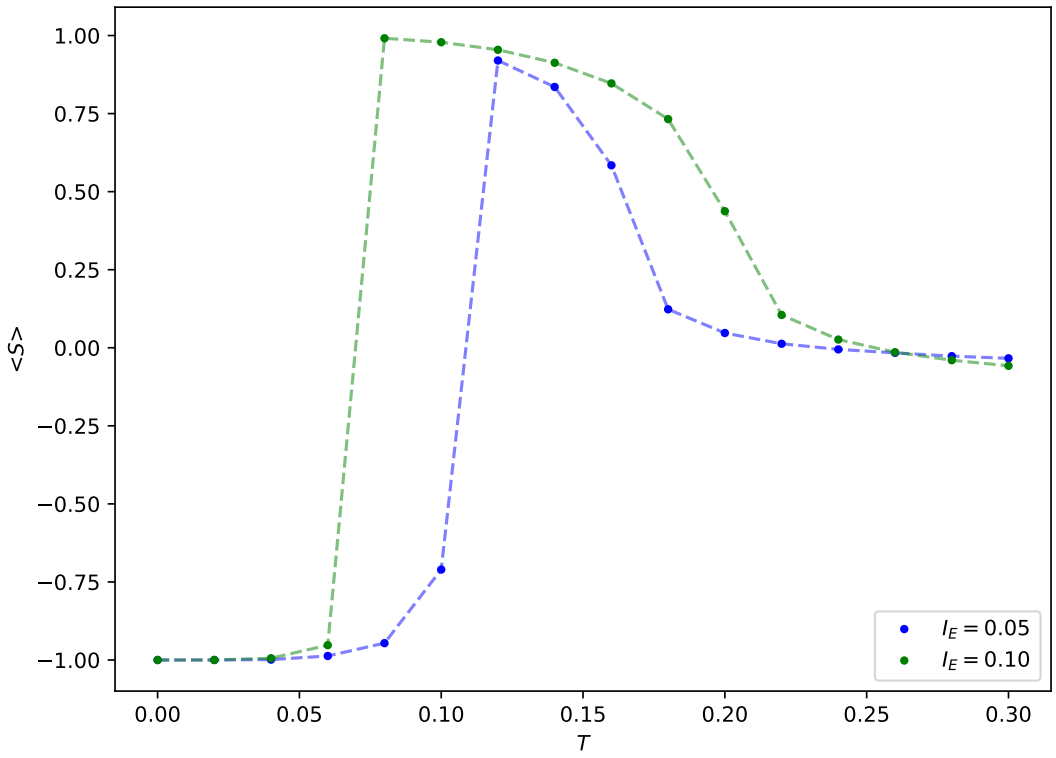


FIGURE 3.14: Relation between $\langle S \rangle$ and the network temperature T for $I_E = 0.05$ (in blue) and $I_E = 0.10$ (in red), starting from a state $\langle S \rangle = -1$. A transition value is observed for $T_t(I_E = 0.05) \approx 0.11$ and $T_t(I_E = 0.10) \approx 0.06$. Values of other parameters are $L = 100$; $L_G = 20$; $p_c = 0.5$; $k_{min} = 8$; $k_{max} = 24$; and $\sigma = 0.3$. Results are averaged over 100 independent simulations. The number of time steps of each simulation is 1000.

opinion. Each data point is obtained by averaging the results of 100 independent simulations. From this we can see that a higher external influence I_E leads to an increase in the critical temperature T_c of the network. The critical transition temperatures are $T_c(I_E = 0) \approx 0.12$, $T_c(I_E = 0.05) \approx 0.16$ and $T_c(I_E = 0.10) \approx 0.18$. Since we are interested in running our model under these conditions, we want to be within the transition regime of the network so that the external influence can be used as the regulating parameter of opinion formation. To better understand how influential I_E is we perform another set of tests but starting from a ferromagnetic phase with sign opposite to I_E. Fig. 3.14 shows just that: we start from $\langle S \rangle = -1$ and act on the network with constant I_E over different temperatures T. This allows us to see the wider range of temperatures where the external influence can sway opinion. We see that for low network temperatures the system stays in it's initial ferromagnetic state which has opposite sign to I_E. This means that for temperatures close to zero, the external influence parameter has no impact on the average opinion of the network - a society under these conditions is too conformist and the status quo is never challenged. As temperature increases we encounter an abrupt transition from $\langle S \rangle = -1$ to $\langle S \rangle = +1$, which indicates the temperature at which I_E becomes a driving force in opinion formation in the network. These occur at $T_t(I_E = 0.05) \approx 0.11$ and $T_t(I_E = 0.10) \approx 0.06$. For network temperatures above this value, the results match Fig. 3.13. Equipped with this information, we set $T = 0.15$ as an acceptable network temperature to simulate a democratic society. Under these conditions, the external influence I_E can be used as a the driving factor in changing electoral outcomes in a society. We ensure that the temperature is not too low in order for a one party state to emerge and that it is not too high that the system is locked into a paramagnetic phase and therefore the electoral outcomes are only subject to thermal fluctuations.

Chapter 4

The DICE model

The DICE model is a so-called Integrated Assessment Model (IAM) used to assess climate policies. It was developed by William Nordhaus, who was awarded the 2018 Nobel Prize in Economics "for integrating climate change into long-run macroeconomic analysis." Nordhaus started working on the DICE model in 1992 [35] and has subsequently improved and updated the model over the years, with its latest release being DICE-2016 [36, 37], which we will make use of here. IAMs combine a simple climate model to a simple macroeconomic model. The aim is to find a sensible climate policy - i.e. the best compromise between the (economic) costs of cutting down CO_2 emissions and the (economic) damages caused by climate change. The model views the economics of climate change from the perspective of neoclassical economic growth theory [38]. In this approach, economies make investments towards future growth thereby reducing present day consumption, in order to increase consumption in the future. DICE extends this approach by including the "natural capital" of the climate system. That is, the concentrations of greenhouse gases are taken to be negative natural capital, and reductions in greenhouse emissions as investments that raise the quantity of natural capital (or reduce the negative capital). By devoting output to emission reductions, economies reduce consumption today but prevent economically harmful effects of climate change and thereby increase consumption in the future.
DICE is primarily designed to optimize policy, but it can also be run as a simple projection model. In both cases, the approach is to maximize an objective function. The objective function refers to the economic well-being (or utility) associated with a path of consumption. There are two control variables in the model (assumed to be determined by human actions/policy). One is the abatement ratio μ, and the other one is the saving rate s (the fraction of the economic production Q used for investment rather than consumption). In principle, humanity can decide on these two variables every time step. For the sake of simplicity, we will only optimize abatement.

4.1 Model Equations

4.1.1 Welfare Function and Discounting

The model tries to find the policy that maximizes the social welfare function W:

$$W = \sum_{t=0}^{T_{max}} U[c(t), L(t)]R(t) \tag{4.1}$$

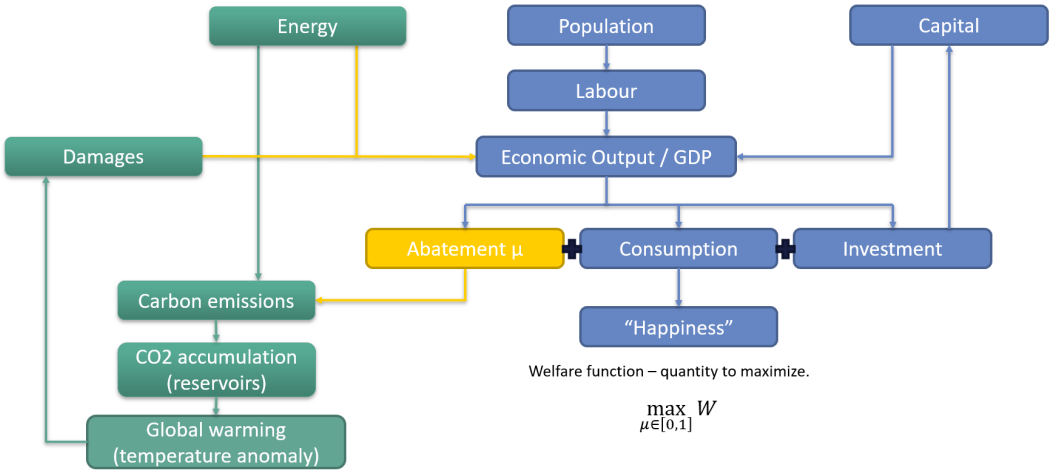


FIGURE 4.1

FIGURE 4.2: Schematic representation of the DICE model. In blue, the economics module; in green, the climate module; and in yellow, the feedback processes linking them together. Adapted from blog article by C. Wieners [39].

where U is the current utility (which we interpret as a measure of "the pleasure the world population enjoys at time t"). U can be written as follows:

$$U[c(t), L(t)] = L(t)\left[\frac{c(t)^{1-\alpha}}{1-\alpha}\right] \tag{4.2}$$

In this equation, the parameter α is a measure of the social valuation of different levels of consumption, which has several interpretations. It represents the curvature of the utility function, the elasticity of the marginal utility of consumption, or the rate of inequality aversion. It effectively measures the extent to which a region is willing to reduce the welfare of high-consumption generations to improve the welfare of low-consumption generations.
If we take (the limit of) $\alpha = 1$, it wields the logarithmic utility function:

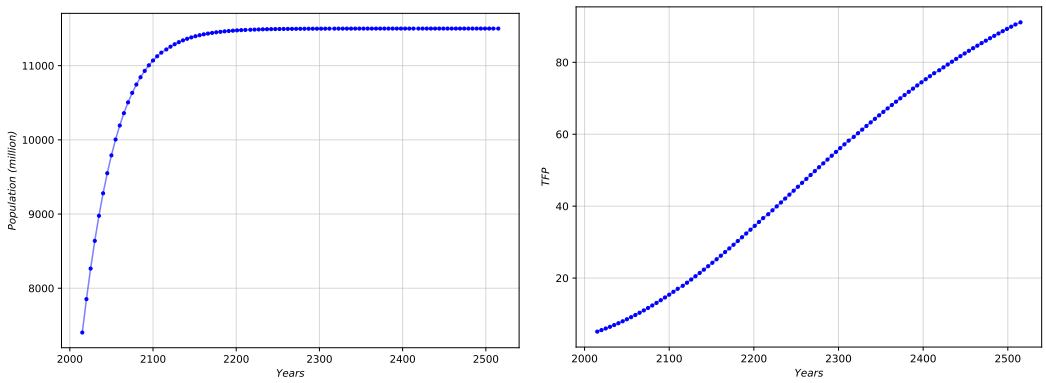
$$U[c(t), L(t)] = L(t)\log(c(t)) \tag{4.3}$$

The logarithmic term always increases with c (the more consumption, the happier), but the increase gets slower when c is high (a certain extra amount of consumption means less to a person who is already consuming much). This means that in principle we prefer to consume the same every year over irregular consumption with the same time-average. Often, α will also be used to represent risk aversion, but these are distinct concepts and should not be confused [40, 41].
Finally, people are supposed to be impatient, and prefer consuming now over consuming later. This is expressed through R (called "discount factor arising from pure time preference"), which decreases with time (later consumption is valued less):

$$R(t) = (1+\rho)^{-t} \tag{4.4}$$

Here ρ is called pure rate of time preference and (if made time dependent) is assumed to decrease slightly in time:

$$\rho(t) = \rho(0)\exp(-g^{\rho}t) \tag{4.5}$$



(a) (b)

FIGURE 4.3: DICE2016 exogenous variables of (a) population and (b) total factor productivity. Time span shown is for the entire 500 years on which the model is run, chosen as such for the purposes of optimization and the time-span of climate change.

where g^{ρ} is the rate of time preference.

The DICE model is rather simplified relative to many models since it assumes a single commodity, which can be used for consumption, investment or abatement. Consumption should be viewed broadly to include not only food and shelter but also non-market environmental amenities and services.

4.1.2 Population Growth

The economic module of DICE is standard to macroeconomic literature. The main difference is the very long time frame that is required for climate-change modelling. While most macroeconomic models run only for a few years, or a few decades, climate-change projections span over more than a century. The result is that many of the projections and assumptions are based on very thin evidence and the validity of DICE in the far future is questionable if not outright wrong.

The population L is assumed to increase, but at a decreasing rate (tending towards a stable value), and is given by:

$$L(t) = L(t-1)\,[1 + g_L(t)] \tag{4.6}$$

where $g_L(t)$ is the population growth rate, which declines with an exogenous rate δ_L, as follows:

$$g_L(t) = g_L(t-1)\frac{1}{1+\delta_L} \tag{4.7}$$

The parameter values are fitted to UN population projections and do not arise from the model itself, with an asymptotic maximum population of 11.5 billion people [42]. This is shown in Fig. 4.3a.

4.1.3 Investment and Economic Output

Every year the population spends a certain ratio of its economic output $Q(t)$ on investment $I(t)$ (the rest is consumed):

$$Q(t) = C(t) + I(t) \tag{4.8}$$

Here $C(t)$ is the total consumption which is defined by the per capita consumption $c(t)$ by:

$$c(t) = \frac{C(t)}{L(t)} \tag{4.9}$$

In absence of climate damage and emission abatement, the economic output is determined by the capital $K(t)$ and the labour (which scales with L):

$$Q(t) = A(t)K(t)^{\gamma}L(t)^{1-\gamma} \tag{4.10}$$

The dynamics of the labour output, called total factor of productivity, $A(t)$ is given

$$A(t) = A(t-1)\left[1 + g_A(t)\right] \tag{4.11}$$

Its development in time is taken from economic projections and is prescribed as:

$$g_A(t) = g_A(t-1)\frac{1}{1+\delta_A} \tag{4.12}$$

The factor δ_A controls the time dependency of $g_A(t)$ and the constant γ determines how economic output depends on capital and labour input. In Fig. 4.3b the total factor productivity can be seen for standard DICE2016 parameters.

The capital K at time step t is determined by the capital at the previous time step $t-1$, investments during the last time period and some annual value loss (machines and so on lose value over time by getting used) called "capital depreciation" (determined by the constant δ_K):

$$K(t) = I(t) - \delta_k K(t-1) \tag{4.13}$$

with the investment I being determined by a saving rate policy:

$$I(t) = s(t)Q(t) \tag{4.14}$$

From here, we must take into account climate damage and policies. Climate change is assumed to reduce Q by a factor which depends on the global mean temperature change with respect to the pre-industrial temperature. In addition, it is assumed that if we reduce CO_2 emissions by a factor of μ (by switching to cleaner energy, not by reducing energy consumption), this induces costs which also reduces Q. Giving us:

$$Q(t) = \left[1 - \Lambda(t)\right] A(t)K(t)^{\gamma}L(t)^{1-\gamma}\frac{1}{1+\Omega(t)} \tag{4.15}$$

$\Lambda(t)$ and $\mu(t)$ are climate related effects to the economy. They represent respectively the climate change damages and the abatement cost (the cost of the reduction of CO_2 emission).

$$\Omega(t) = \phi_1 T_{AT}(t) + \phi_2 T_{AT}(t)^2 \tag{4.16}$$

$$\Lambda(t) = \theta_1 \mu(t)^{\theta_2} \tag{4.17}$$

The damage equation (4.16) quantifies the economic impacts of climate change, which is a delicate issue in climate-change economics. These estimates are essential for making reasonable decisions about the appropriate balance between costly emissions reductions and the damages induced by climate change. However, providing reliable estimates of these damages over such long time spans has proven to

be extremely challenging. Nordhaus revisits this multiple times and this form of the damage function is taken with consultation of Tol's surveys [43, 44]. Ultimately, it is extremely difficult to quantify damages and account for a plethora of factors, such as: economic value of losses from biodiversity, ocean acidification, political reactions, extreme events (sea-level rise, changes in ocean circulation and tipping points in the climate system), impacts that are inherently difficult to model (catastrophic events and very long term warming) and uncertainty associated with virtually all components of the model.

Substitution from fossil fuels to renewable energy takes place over time as fossil fuels become more expensive, either due to resource exhaustion or because policies are taken to limit carbon emissions. Renewable energies also get less expensive over time as supply chains, economies of scale and improvements in technology take place. DICE also includes a backstop technology, which is a technology that can replace all fossil fuels. This essentially leads to the possibility for negative emissions (carbon capture). The backstop technology could be one that removes carbon from the atmosphere directly or an all-purpose environmentally benign zero-carbon energy technology (whatever that might be). The price of this technology is assumed to be initially high and to then decline over time with technological improvements.

4.1.4 Carbon Emissions

Global annual industrial CO_2 emissions E_{ind} depend on how strong our economic activity is and how carbon-efficient our energy is:

$$E_{ind}(t) = \sigma(t)\left[1 - \mu(t)\right] A(t) K(t)^{\gamma} L(t)^{1-\gamma} \tag{4.18}$$

where μ is again the carbon abatement, and σ scales the CO_2 emissions per economic output in absence of climate policy. The parameter σ is prescribed and decreases slowly (even without climate change we need less carbon per energy because we learn to make more efficient power plants for cost reasons):

$$\sigma(t) = \sigma(t-1).\left[1 - g_{\sigma}(t)\right] \tag{4.19}$$

The time evolution of g_{σ} is given by:

$$g_{\sigma}(t) = g_{\sigma}(t-1).\frac{1}{1+\delta_{\sigma}} \tag{4.20}$$

Land use change also leads to CO_2 emissions $E_{land}(t)$ which are prescribed and regulated by δ_1:

$$E_{land}(t) = E_{land}(0)(1-\delta_1)^t \tag{4.21}$$

The total CO_2 emissions per year are then given by:

$$E(t) = E_{ind}(t) + E_{land}(t) \tag{4.22}$$

We also set a limit of carbon emission as a constraint:

$$CCum \geq \sum_{t=0}^{T_{max}} E_{ind}(t) \tag{4.23}$$

Carbon fuels are limited in supply, with a total limit set at 6000 billion tons of carbon. This constraint is non binding for a baseline projection.

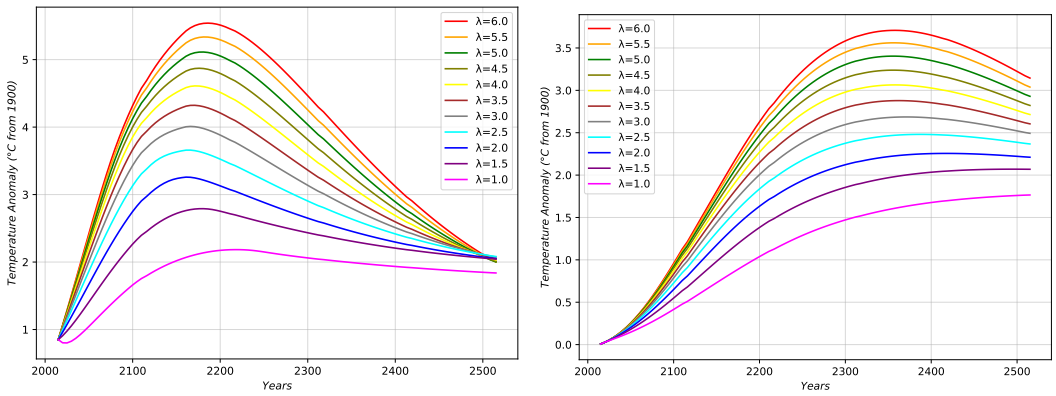


FIGURE 4.4: Temperature anomaly under the optimal climate policy for different values of climate sensitivity λ. The maximum value of each curve corresponds to the selected peak temperature projected for that optimal path. All optimization curves shown correspond to initial settings at the year 2015. (a) Atmospheric temperature anomaly. (b) Average ocean temperature anomaly.

4.1.5 Geophysical Equations

The climate model of DICE includes several geophysical relationships that are linked to the economy with the different feedbacks that lead to man-made climate change. These include the carbon cycle, radiative forcing, climate-economy coupling, and a climate-damages relationship. Since DICE is an IAM, the modules operate in an integrated fashion rather than taking variables as exogenous inputs from other models or assumptions. One needs to simplify the inherently complex dynamics of the climate system into a small number of equations that can be used in an integrated economic-geophysical model.

In DICE, the only greenhouse gas that is subject to abatement is industrial CO_2. This is simply because CO_2 is the major contributor to global warming and that other greenhouse gases are likely to be controlled in different ways. Other greenhouse gases are included as an added exogenous term in radiative forcing; these include primarily CO_2 emissions from land-use changes, other well-mixed greenhouse gases and aerosols.

The emitted carbon first gets into the atmosphere (AT) , but from there, a certain part is absorbed by an "upper" (UP) reservoir (land biosphere and top layers of the ocean), from here some part is transferred into the lower (LO) ocean. On the other hand, the lower ocean might also release carbon to the upper ocean and the upper ocean to the atmosphere. M_X is the mass of carbon (in Gigatons of C) in reservoir X and ϕ_{ij} are positive coefficients determining carbon exchange between reservoirs:

$$M_{AT}(t) = E(t) + \phi_{11} M_{AT}(t-1) + \phi_{21} M_{UP}(t-1) \tag{4.24}$$

which means that atmospheric CO_2 content depends on its previous value, and emissions in the previous time step. In addition, the atmosphere gains $\phi_{21} M_{UP}(t-1)$ from the upper ocean. The atmosphere does not communicate directly with the lower ocean.

$$M_{UP}(t) = \phi_{12} M_{AT}(t-1) + \phi_{22} M_{UP}(t-1) + \phi_{32} M_{LO}(t-1) \tag{4.25}$$

$$M_{LO}(t) = \phi_{23} M_{UP}(t-1) + \phi_{33} M_{LO}(t-1) \tag{4.26}$$

The mixing between the deep oceans and other reservoirs is extremely slow and the deep oceans provide a large sink for carbon in the long run. The carbon which is present in the atmosphere leads to radiative forcing F (i.e. blocks some long-wave radiation which would otherwise escape into space). However, there is also radiative forcing not related to CO_2 (e.g. from other greenhouse gases). These contributions are denoted by $O(t)$ and are prescribed.

$$F(t) = \eta \left\{ \ln \left[\frac{M_{AT}(t)}{M_{AT}(1750)} \right] \right\} + O(t) \tag{4.27}$$

where η is a constant (the radiative forcing arising from doubling M_{AT} if all other greenhouse gases remain at pre-industrial levels), and $M_{AT}(1750)$ the pre-industrial value of atmospheric carbon.

The radiative forcing ensures that energy which otherwise would escape into space is retained in the climate system, warming the earth (implicitly it is assumed that the pre-industrial state was in equilibrium). This energy can end up in the surface reservoir (land, atmosphere, upper ocean), which has a relatively low heat capacity and reacts quickly to radiative forcing, with temperature changes T_X, with X being the carbon reservoir. We will from now on refer to it as the "temperature anomaly", since this quantity corresponds to the difference in temperature with respect to pre-industrial levels. Part of the heat is slowly transferred to the lower ocean, causing temperature changes there (T_{LO}). The equations are:

$$T_{AT}(t) = T_{AT}(t-1) + \xi_1 \{ F(t) - \xi_2 T_{AT}(t-1) - \xi_3 \left[T_{AT}(t-1) - T_{LO}(t-1) \right] \} \tag{4.28}$$

$$T_{LO}(t) = T_{LO}(t-1) + \phi_4 \{ T_{AT}(t-1) - T_{LO}(t-1) \} \tag{4.29}$$

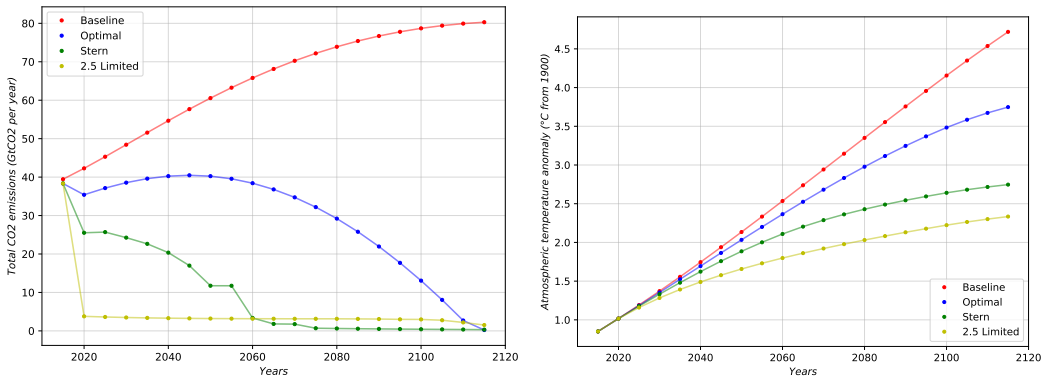
where ξ_3 governs the heat transfer from the lower to the upper ocean and ξ_4 the heat transfer from the upper to the lower ocean. Note that in equilibrium ($T(t) = T(t-1) = T_{LO}(t) = T_{LO}(t-1)$) the equilibrium temperature T_{eq} becomes $F(t)/\xi_2$, hence η/ξ_2 is the “climate sensitivity”, i.e. temperature change (w.r.t. pre-industrial value) when doubling the carbon concentration w.r.t pre-industrial levels. This is governed by the relation:

$$\Delta T_{AT}(t) = \frac{\Delta F(t)}{\xi_2} \tag{4.30}$$

The equilibrium climate sensitivity is an important concept in climate physics and DICE-2016 takes it as being 3.1 °C for an equilibrium CO_2 doubling. The value of climate sensitivity is riddled with large uncertainties despite it being well document in literature and is subject to contentious debate, but an extended discussion of this topic is beyond the scope of this project (for a detailed discussion see [45, 46]).

4.2 Model Scenarios

We now present and briefly discuss some well established scenarios that result from DICE. Nordhaus focuses his attention primarily on the economic impacts, but we are more concerned with the average temperature anomaly and how it can be regulated through policy. Therefore we shown the carbon emissions (Fig. 4.5a) and average temperature anomaly (Fig. 4.5b) for the next century and for four distinct scenarios as calculated by our implementation of DICE-2016 (these are in agreement with literature [37]). The baseline case (in red) represents the outcome of market and policy factors given current circumstances. It is an attempt to project the levels and growth



(a) (b)

FIGURE 4.5: (a) Yearly CO_2 emission total and (b) atmospheric temperature anomaly.

of major economic and environmental variables as they would occur with current climate-change policies. It does not make any case for the social desirability of the distribution of incomes over time of the existing conditions. Note that this does not mean that no abatement policy is being taken, in fact it takes into account a rough estimate for μ based on current efforts to mitigate climate change. This approach is standard for forecasting, say of government budgets, and is more appropriate for a world of evolving climate policies - which is what we will do once we couple DICE to our democratic network. The baseline case, nonetheless requires an estimation of emission intensities and therefore multiple emission scenarios can be taken into account. Under current projections, carbon emissions will continue to increase as the economy grows over the next century. Transition to renewable energies is minimal and this is projected to lead to a large average temperature anomaly of 4.5°C by year 2110. One can't blame the relentlessness of climate activist in pressuring world leaders to act when asked to dwell on these grim "business as usual" projections. For the optimal policy scenario (in blue), abatement policies maximize economic welfare, with full participation by all world governments starting in 2015 and without any climate constraints. This scenario assumes the most efficient climate-change policies; in this context, efficiency involves a balancing of the present value of the costs of abatement and the present value of the benefits of reducing climate damages. This is of course unrealistic, but this scenario provides an efficiency benchmark against which other policies can be compared to. A similar optimization curve can be calculated depending on the ideology of political parties, should they make use DICE to inform their abatement policy decisions. Under this scenario, carbon emissions are reduced over the next century until we reach a carbon neutral economy by year 2115. The implementation of abatement policies lead to a steady transition towards renewable energies. However, the temperature anomaly still reaches a somewhat large value at 3.5°C by year 2100. This is a consequence of the continued rate of carbon emissions and also the lag of the climate system into reaching a state of equilibrium and reacting to the CO_2 that we already emitted in past years, whose consequences are only now starting to be felt. We also showcase two other scenarios that are prevalent in the literature: the Stern Review scenario [47] and the Paris Agreement's limit of 2.5°C temperature increase scenario [48]. The Stern Review advocates using very low discount rates for climate-change policy. This was implemented using a time discount rate of 0.1% per year and a consumption elasticity of 1. This leads to low real interest rates and generally to higher carbon prices and emissions control rates. Stern argues this with moral concerns, stating that one needs to take

the well being of future generations into account. Since climate change is such a long scale problem, it will disproportionally affect future societies. Under this scenario (in green), stringent abatement policies are introduced as we discount towards the future more than Nordhaus' standard settings and prefer to consume less at present time. It leads to a quick transition into a carbon neutral economy, but it has a harsh impact on economic growth. As for the last scenario, the 2.5-limited scenario (in yellow), we show just how dramatic emissions would have to be reduced in order for the very ambitious targets of the Paris Climate Agreement to be met. Here, the optimization of abatement is subject to the constraint that global temperature does not exceed 2.5°C above the 1900 average. This "temperature-limited" scenario is a variant of the optimal scenario that builds in a precautionary constraint that a specific temperature increase is not exceeded (the upper limit of the Paris Agreement is set at 3°C).

Chapter 5

DICE-Voter Coupling

In order to incorporate voter behaviour into the DICE model, we have to develop a coupling mechanism between the social network that we have previously constructed to simulate opinion dynamics and the multiple iterations of DICE. The time step of DICE is typically chosen to be 5 years, which can be taken as a full election cycle for a democratic society. This means that at each iteration of DICE we will have to consult the social network and the resulting opinion average $\langle S \rangle$. The opinion average then reflects which political party has won the election and stays in power for the duration of that election cycle, and they get to implement their policies regarding the climate. Once elected, the political party will select an optimal value of abatement which is obtained by an optimization of DICE that takes into account their particular world view regarding climate change and the economy. We therefore make a distinction between the optimized, party-specific DICE model and the "real-DICE", which is the real-time model where election cycles occur. A schematic representation of this is shown in Fig. 5.1.

Here arises the challenge of finding an adequate way to bridge the information obtained by the climate model to the perceived idea that a society has about it. Essentially, we want to translate the DICE output for a particular election cycle into the external influence parameter I_E, and use that to influence voter behaviour.

5.1 Political Parties

When considering agents with respect to climate change, Geisendorf [49] suggested two extreme opposites he labelled “Individualists” and “Egalitarian” agents. Individualists believe in the power of free market forces and a great resilience of nature. Nature tends to equilibria, which, after perturbations, reinstall themselves. They believe the climate-economic system will fix itself and economic activity should not be restricted. In contrast, Egalitarians are fundamental environmentalists and very risk-averse. Imbalances in the natural equilibrium will lead to disaster and thus requests to prevent any strong impact on nature. They’d rather live on a very basic, but equally distributed level of wealth than risk to disturb nature with our capitalist economy. Egalitarians opt for zero growth and high environmental protection.

These are in stark contrast with one another, and it’s perfectly possible to attempt to describe the intentions of these two agents. However, for such extreme views it is not so difficult to decide on how to construct the parameters of the DICE model coupling since, if in power, the Individualists would easily simply choose not to optimize abatement policy and run with zero abatement. Egalitarians would push for abatement policies that could meet rigorous temperature anomaly constraints regardless of the consequences for the economy.

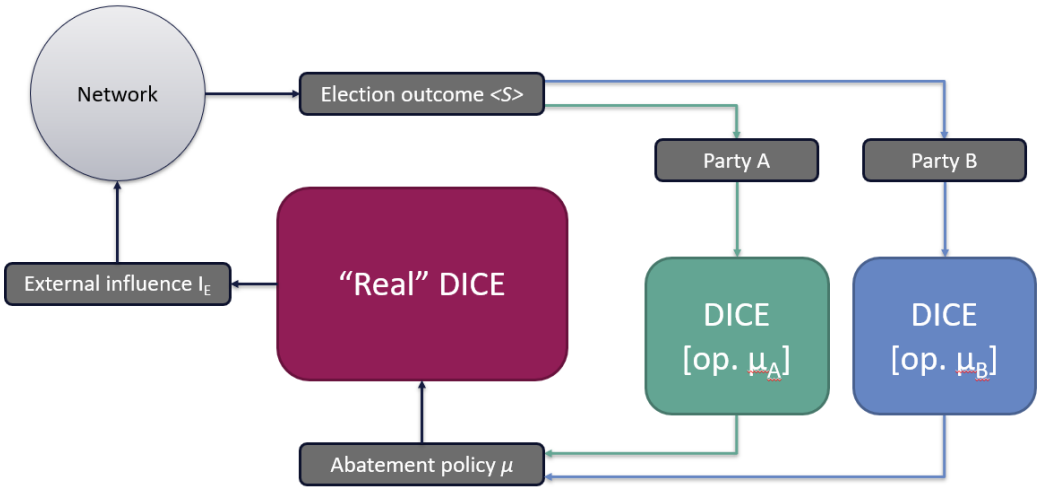


FIGURE 5.1: Schematic representation of the coupled DICE model. The real-DICE projects the outcome over the course of one century. The coupling equation conveys this yearly outcome to the network in the form of the external influence I_E, resulting in party A or B winning the election. Each party optimizes the policy outcome based on the economic and climate variables at the year they are elected and implements said policy over the course of the next election cycle.

We consider two distinct political parties, one is the "Lukewarmers"[1] and the other being the "Greens". The former think that climate change is a problem, but not so big or urgent one; the latter think it's a big problem, but not to the point as to hinder economic growth (at least if it can be avoided). In this way, both parties are somewhat less extreme than the ones proposed by Geisendorf.

In order to build an appropriate coupling equation that translates the results of DICE into an external influence onto the network opinion, we will go through a rather extensive list of variables that these parties could disagree about.

1. **Rate of Pure Time Preference:** Disagreements on the rate of pure time preference are not new [47]. The Nordhaus standard value, which is based on observations of the market and is considered by many as unethically high, is easily linked to the Lukewarmers. The Greens should aim for a rate of pure time preference that is as low as possible without crashing the model - DICE suffers from these problems when taking low discount rates.

2. **Damage Function:** Agreement on what form the damage function should take could be an easy point of divergence between parties. The main issue is that consensus on what that form should be is not clear and therefore, all sorts of damage functions could be chosen either way, as shown by Weitzman [50] [51]. Lukewarmers could easily take Nordhaus' standard damage function, which most scientists agree to be on the mild side. The Greens could take, for example: Nordhaus$\times$3, or a threshold damage function, where the damage gets very serious as soon as a particular threshold is reached [52]. Weitzman's damage function is not very suitable, because his proposed damage function only becomes very drastic from about 4K warming; below 3K, it very much resembles Nordhaus'.

[1] One who believes that climate change is due to human activity but who does not think it is a serious problem.

3. **Climate Sensitivity:**

   (a) Simple; The two parties could hold different views on the climate sensitivity. The main disadvantage is that climate sensitivity is scientifically well researched [45] - uncertainty is big, admittedly, but no party can claim that they are sure of its value. Therefore, this is not realistic behaviour. One can still argue that both parties have the scientific information on climate sensitivity, but they interpret it differently: The Lukewarmers would go for the typical value (3K/doubling CO_2), while the Greens could plan for the worst case scenario (or at least a rather more pessimistic one), and optimize as if the climate sensitivity is 5K/doubling CO_2.

   (b) Risk Aversion; Alternatively, one can let both parties acknowledge that there is uncertainty, and let them optimise expected welfare (depending on the probability density function of the climate sensitivity). However, the Green party applies risk aversion and the Lukewarmers not. Interestingly, if there is no risk aversion, then the policy under uncertain climate sensitivity (optimising expected welfare) is very similar to what one gets when optimising the ordinary model with the expected climate sensitivity.

   (c) Bayesian Learning; Parties might initially have different beliefs about climate sensitivity, but learn the true value gradually through measuring temperature and comparing it to past carbon emissions. A mechanism like this would be an interesting addition that could better account for discrepancies between optimized DICE and the "real" DICE.

4. **Exogenous Carbon Intensity Improvement:** In DICE, the carbon intensity (that is, the carbon emissions in the absence of policy) decreases over time, about 0.5%/year. The Lukewarmers could use a stronger decrease than the Greens (e.g. 1%/year vs 0.25%/year), i.e. the Lukewarmers believe that the climate problem to some extent solves itself through technological progress.

5. **Cost of Abatement:**

   (a) Parametric; In such case, the Lukewarmers would consider abatement to be costly, as is done in the standard DICE. The Greens however, think it is not too bad - around a third of the standard DICE. DICE is probably rather on the high side when it comes to abatement costs, especially in the long run [53].

   (b) Structural; The Lukewarmers could believe that abatement will be costly forever (costs depend on μ) and the Green party thinks that abatement is investment-like, i.e. costs are transitional (depend on $\dot{\mu}$). (again, see [53]). This will make Lukewarmers want to postpone abatement, whereas Greens believe that we should tackle the transition quickly and then live happily ever after. This has the following caveat: if costs are partly transitional, then it does not hold anymore that the marginal abatement cost (the cost to abate 1 extra ton of carbon - i.e. the desired tax value) equals the social cost of carbon [37]. The reason is that with transitional costs, some expenditure we do today (e.g. building a solar cell) will also save carbon tomorrow. So in fact a lower carbon tax suffices to bring about the necessary abatement.

5.2 The Coupling Equation

Voter's behaviour concerning climate change could be influenced by three main points: discontent about current taxation, discontent about current climate damages, and a worry about (expected) future temperature change. This yields an equation of the form:

$$\bar{I}_E(t) = \bar{\alpha} D(t) - X(t) + \bar{\omega} F(\hat{T}) \tag{5.1}$$

where $D(t)$ is the current climate damage as given by the DICE damage function and $X(t)$ is the carbon tax or abatement cost. Both are normalised so that the payment is per inhabitant. $\hat{T}$ is the expected peak temperature, which we will discuss further below). F is some function of temperature (could be identity, but could also for example equal the damage function). We call $\bar{\alpha}$ the "awareness parameter", which is some measure of how much people understand that the damage is caused by climate change, and $\bar{\omega}$ is called the "worry parameter", which captures how much people take future threats into account. The value $\bar{I}_E$ is related to the external influence parameter of the network I_E by a scaling factor ζ such that $I_E = \zeta \bar{I}_E$. The scaling factor must be calibrated to the specific network that we use and is chosen in a such a way that the network maintains its paramagnetic state - that is, the voter opinion only fluctuates around $\langle S \rangle = 0$ and dramatic election results (landslides) don't occur. A higher $\bar{I}_E$ translates into a higher acceptance for stringent climate policy.

The tax $X(t)$ is simply the carbon tax raised by the parties. This would always be the parties social cost of carbon (SSC) along their optimal path - except if a party assumes abatement costs to be transitional.

We can expect $\bar{\alpha} D(t) - X(t)$ to always be negative, at least in the first few decades, because $D(t)$ only takes into account current damages for that specific time step, which is initially low, while $X(t)$ takes future damages into account, even when discounted. A population that is then unable to look into the future will always oppose climate policy. In order to counteract this, we make use of $\hat{T}$ which is a measure of expected future climate change, specifically it corresponds to the future peak temperature under the current abatement policy being implemented.

However, the question arises as to how this peak temperature expectation is made; that is, if it is obtained by the party-specific DICE optimization or by the real-time DICE. It could be that the public uses the real-DICE model to infer expected future temperatures - maybe informed by what we could consider to be politically unmotivated scientists - but it could also be that the political parties propagate their own predictions of future temperatures as determined by their optimization scheme. If the parties have a different view on climate sensitivity (or use a different F, e.g. because of different damage functions), then probably their own predicted $F(\hat{T})$ turn out to be more similar than reality. This is because the Lukewarmers would do a mild climate policy (which should lead to high future climate change) but at the same time they propose that climate (damage) does not react strongly to emissions. The Greens, on the other hand, do a stringent abatement policy, so they will keep climate change lower, but at the same time work with the assumption of a sensitive climate. Hence, the voters have a harder time distinguishing the parties' future climate impact, if they strongly believe the parties' own predictions.

Mathematically, one could in fact formulate (at least if F is the identity)

$$\hat{T}(\mu) = \gamma_{op} \hat{T}_{op}(\mu) + \gamma_{gov} \hat{T}_{gov}(\mu) + \gamma_{sci} \hat{T}_{sci}(\mu) \tag{5.2}$$

where μ is the current abatement policy, and $\hat{T}_x$ is the peak temperature expected by group x - the current opposition party, the current governing party and the scientists who are politically neutral). The γ are all parameters in $[0, 1]$ such that $\gamma_{op} + \gamma_{gov} + \gamma_{sci} = 1$ and $\gamma_{op} < \gamma_{gov}$. In fact, γ_{gov} and γ_{op} could very well be made time dependent. It would be affected by the time that the governing party is already in power - government propaganda takes some time to sink in. If the two parties (and the scientists) not only have different peak temperature $\hat{T}$ but also different $F(\hat{T})$, then the above equation takes a more general form given as

$$F(\mu) = \gamma_{op} F_{op}(\hat{T}_{op}(\mu))) + \gamma_{gov} F_{gov}(\hat{T}_{gov}(\mu)) + \gamma_{sci} F_{sci}(\hat{T}_{sci}(\mu)) \tag{5.3}$$

5.3 Coupling Regimes

We choose a relatively small network, with somewhat realistic parameters, to be coupled to the DICE model. The chosen settings are $L = 200$; $L_G = 20$; $k_{min} = 10$; $k_{max} = 100$; $p_c = 0.5$; $\sigma = 0.3$ and $T = 0.15$. We select climate sensitivity as the distinguishing element between political parties due to its simplicity. Each party runs their DICE optimization schemes using the starting conditions of the current election year, with the Green party taking climate sensitivity as being 5K/doubling CO_2 and the Lukewarmer party using 3K/doubling CO_2.

5.3.1 Scale Factor

In order to properly couple the output of DICE and port it over to the network, we have to find the appropriate scale for the coupling equation. In order to do this, we investigate the effect of ζ onto the coupled DICE model for a simple case where we take a form of Eq. (5.1) with $\bar{\alpha} = 1$ and $\bar{\omega} = 0$ such that:

$$I_E = \zeta \bar{I}_E(t) = D(t) - X(t) \tag{5.4}$$

The damages and tax contributions are normalized with respect to their initial model values in order to make the coupling a dimensionless quantity:

$$\bar{I}_E(t) = \frac{D(t)}{D(0)} - \frac{X(t)}{X(0)} \tag{5.5}$$

To find a regime where political balance is maintained, we must ensure that the coupling equation does not bridge over to a range of values for the external influence parameter I_E that cause an abrupt phase transition from the starting paramagnetic state of the network to a ferromagnetic state. We expect the coupling equation to cause party shifts over the election cycles, but not to observe dramatic majority election results. A simple demonstration of this can be seen in Fig. 5.2a, which shows the external influence parameter that comes out of the coupled DICE model for the first century. Abrupt shifts in I_E over successive iterations are to be expected due to the nature of the coupling equation, as the influence of climate policy leads to an increase in the carbon tax in the later years of the model. The election results that arise from the I_E are shown in Fig. 5.2b, and we can clearly see that if the external influence parameter becomes too high it leads to large election majorities. As we are mostly interested in regimes where no party wins the election by huge margins, Figs. 5.2c and 5.2d show the results for lower values of ζ. It is not difficult to see that an increase in the scale factor ζ will lead to the external influence parameter taking

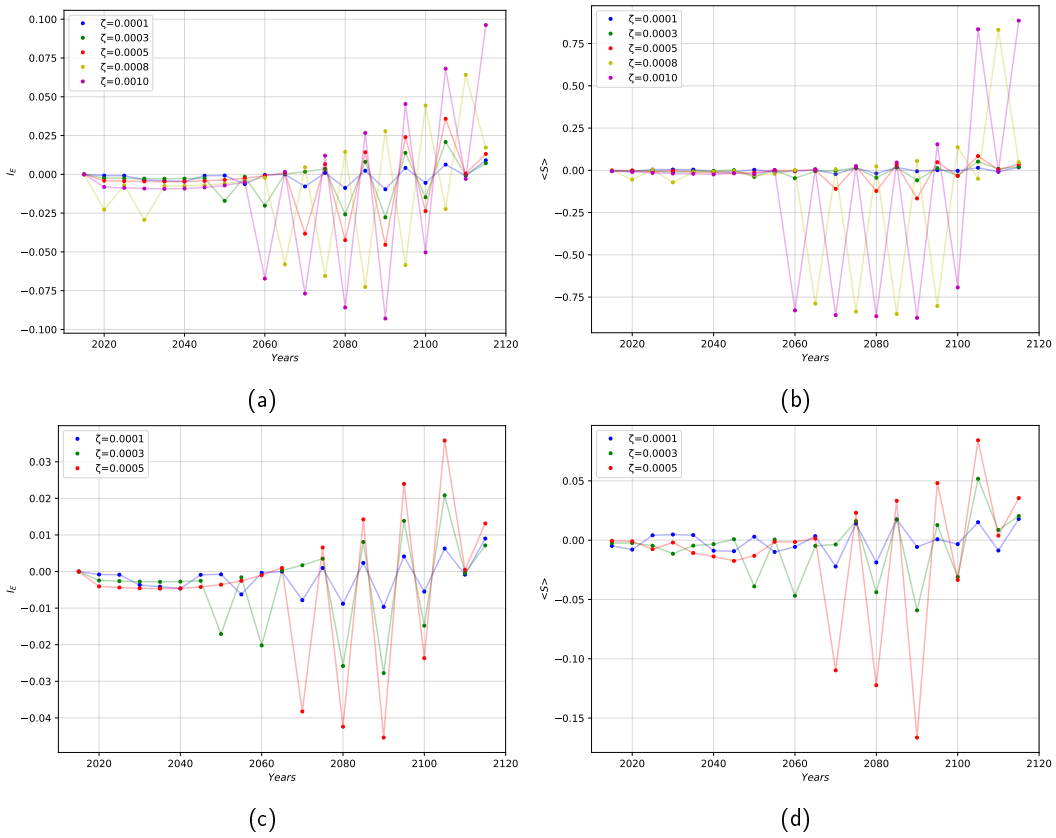


FIGURE 5.2: Output of the coupled DICE model for different scale factors ζ. (a) The scaled external influenced parameter I_E. (b) The resulting opinion average of the network, translating to the political party selection. (c)(d) Same as the previous plots, but showing only the paramagnetic regimes - where dramatic election landslides do not occur.

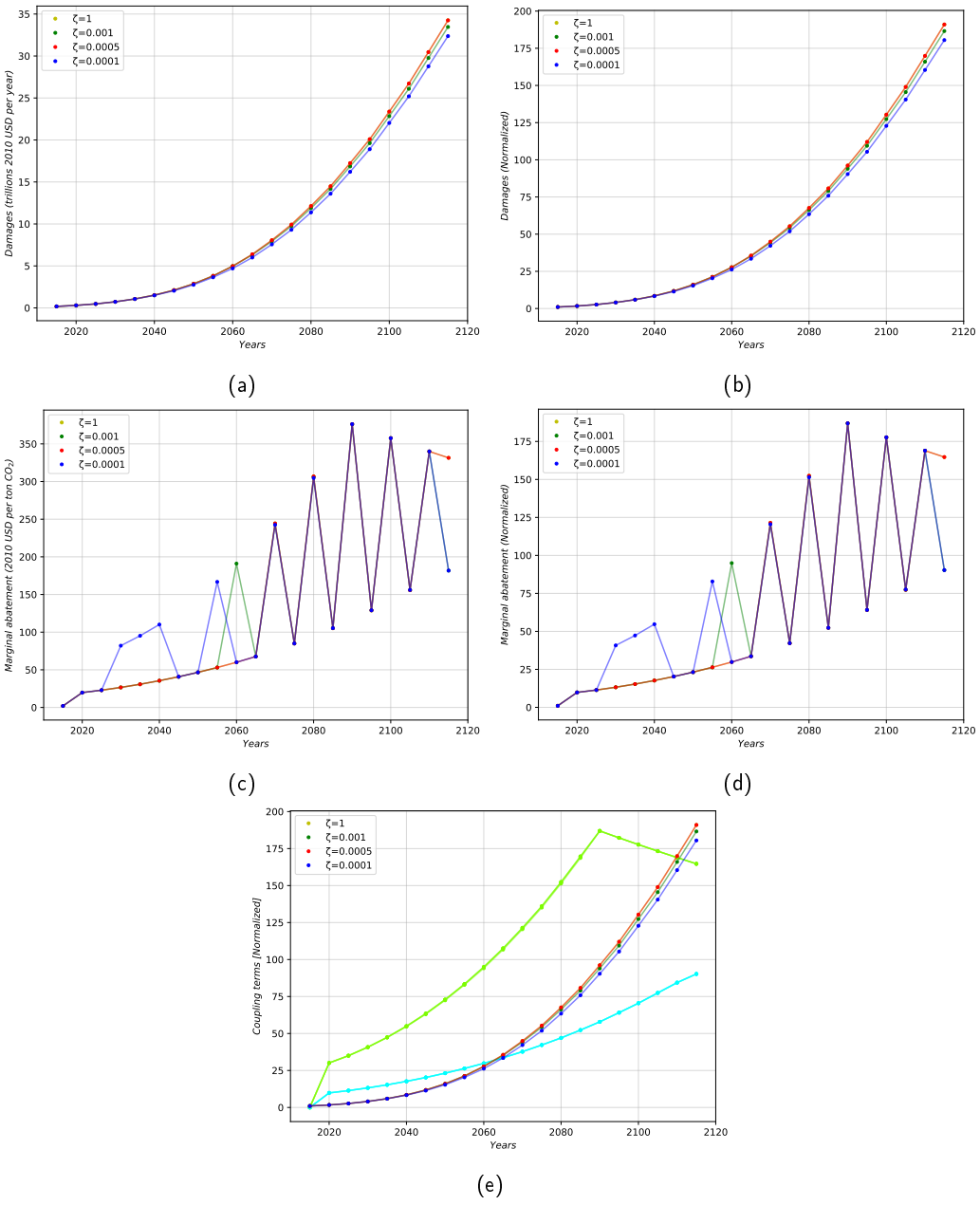


FIGURE 5.3: The behaviour of the main terms of the coupling equation for different scale factors ζ. (a) Time evolution of climate-induced damages. (b) Normalized (dimensionless) damages. (c) Time evolution of the climate tax (here the marginal abatement cost). (d) Normalized tax. (e) Comparison of the range of the coupling terms - Green party optimal path in lime green (upper boundary) and Lukewarmer party optimal path in light blue (lower boundary).

control of the value of the local field h_{ij} as given in Eq. (3.3). This is consistent with the phase transitions from a paramagnetic to ferromagnetic state seen in Fig. 3.13.

It is also important to observe how the different terms of the coupling equation behave over time. The contribution of climate damages is the most stable part of the coupling equation as we can see in Fig. 5.3a and 5.3b. Since climate damages are determined by the "real" DICE, there is very little change in the damage curves for different scale factor ζ. The damage function is a function of the atmospheric temperature anomaly given by Eq. (4.16), therefore, the impact of each election cycle and subsequent party/policy shifts won't be significant in the short term. Damages do increase considerably over time, a consequence of rising atmospheric temperature (Fig. 4.4a). Our "real" DICE runs with climate sensitivity $\lambda = 3.1$ and the projected temperature anomaly is roughly equivalent to the grey curve. Although, we point out that the temperature curves in Fig. 4.4 correspond to optimal paths. The "real" DICE merely projects based on the governing party climate policy, it does not optimize.

The tax term in the coupling equation ends up being the main factor that influences party selection. It fluctuates considerably between political parties as can be seen in Fig. 5.3c and 5.3d, a consequence of the different optimal policy paths calculated by each party. This is easily seen in the tax curves - the upper values follow the Green party's optimal path ($\lambda = 5$), while the lower values follow the Lukewarmers' path ($\lambda = 2$). When putting these two contributions together into the coupling equation, we perform a normalization of the results in order to get I_E as a dimensionless quantity and to ensure that both terms operate within the same range. This is a simple way to compare the evolution of both terms, but it should be taken into account that for the coupling equation to have a meaningful interpretation, the units of both terms must be made to match Nordhaus' model (which we will later do).

A major consequence of how we construct the coupling and of the dominating effect of the tax term on the external influence parameter I_E is that the coupling equation is effectively bound between the tax curves of each party (Fig. 5.3e). The optimal curves for each party were shown not to be meaningfully affected by the time evolution of our "real" DICE model. In other words, the optimal curves that result from the initial conditions at the year 2015 are the same as if we run the party optimization scheme for initial conditions taken every 5 years (differences are minimal and have no relevant impact on the resulting abatement policy μ). This means we can streamline the model by removing the optimization altogether (it only needs to be done once) and the resulting climate policy is always taken from the specific bounds at the respective year a party wins an election. Note that this is only true for an unchanged ideology. Should the political parties adjust their beliefs on the go, then the resulting tax curves would differ.

5.3.2 Awareness Parameter

A significant issue that we have encountered is associated with the contribution of the climate damages term. Our dimensionless coupling equation suffers from a clear dominance of the tax term on the electoral outcome, by being the main factor determining the external influence parameter I_E. In order to balance out this effect, we resort to a control parameter $\bar{\alpha}$, called the awareness parameter. This is introduced in order to mediate how much a population cares about the impact of current climate damages. The awareness parameter $\bar{\alpha}$ can be a constant or it can be chosen to be time depend $\bar{\alpha}(t)$, although its functional form is subject to debate. Looking at the dimensionless normalization of the coupling equation that we previously introduced, we

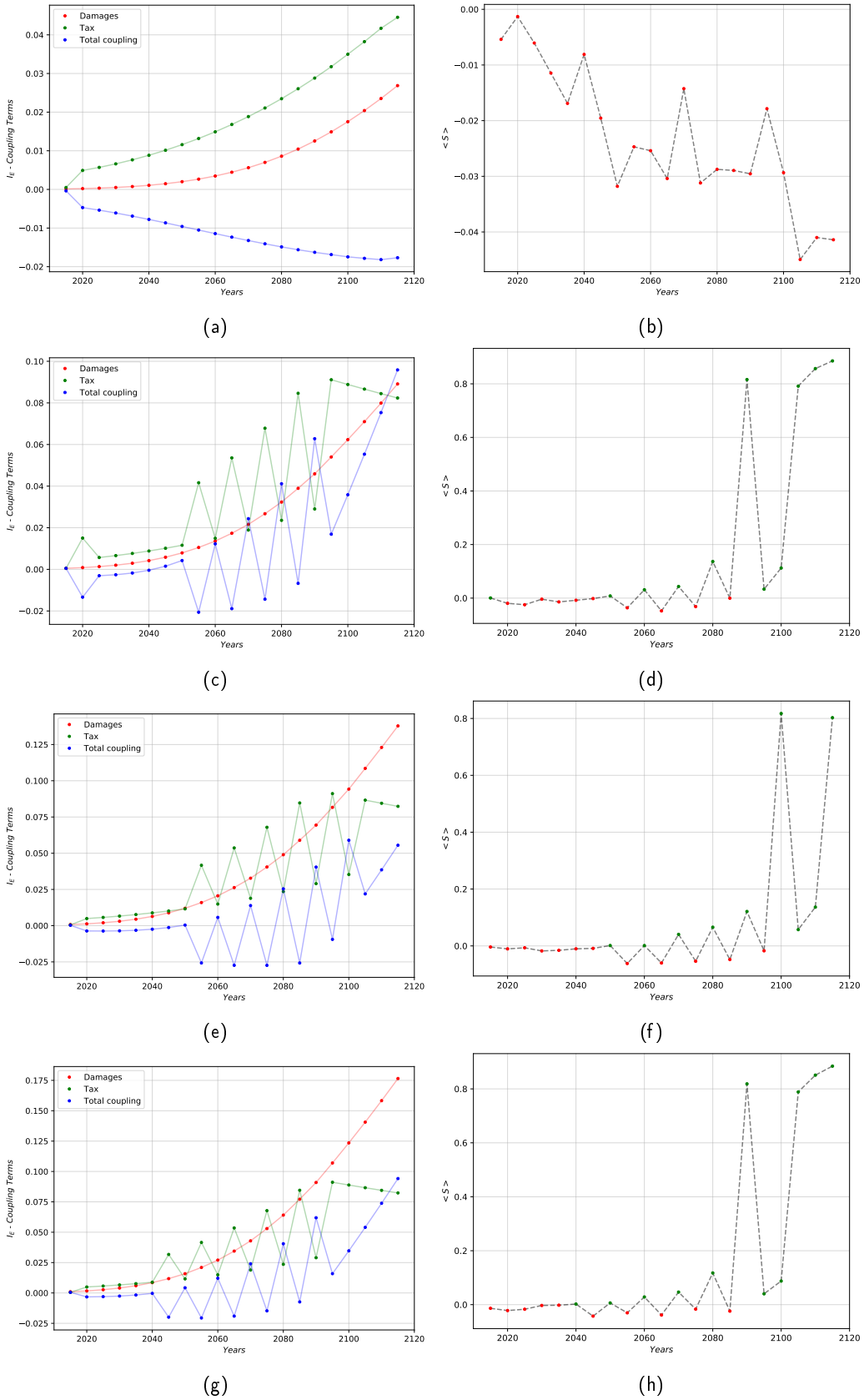


FIGURE 5.4: Effect of awareness $\bar{\alpha}$ on a dimensionless normalization of the coupling equation. The scale parameter is set at $\zeta = 0.0005$. On the left, the resulting external influence parameter I_E and its respective tax and damage contributions. On the right, the resulting opinion average. (a)(b) $\bar{\alpha} = 0.25$ (c)(d) $\bar{\alpha} = 1.0$ (e)(f) $\bar{\alpha} = 1.5$ (g)(h) $\bar{\alpha} = 2.0$.

can see how $\bar{\alpha}$ (chosen as a constant value) affects the electoral outcome. This can be seen in Fig. 5.4, with the external influence I_E and its respective terms shown on the plots on the right and the consequential opinion average being shown on the plots to the left. We can see that for $\bar{\alpha} < 1$, the contribution of damages to the coupling turns out to be irrelevant and leads to an outcome favourable towards the Lukewarmers (Fig. 5.4b), by guaranteeing that the tax term will always control the value of I_E (Fig. 5.4a). This, of course, is to be expected given what was previously established with respect to the climate term contribution. The awareness parameter must be $\bar{\alpha} \geq 1$ in order for it to have any meaningful impact on the coupling equation. Increasing the awareness parameter will amplify the contribution of climate damages to the coupling equation and push the result in favour of the Greens. The balance of power is maintained for most of the simulation, but eventually the tax term will reach its highest value and lose relevance over time due diminishing returns. Since climate damages continue to climb even after the tax term is capped, we see phase transitions towards a ferromagnetic state for later years, also to be expected. Its important to point out that for the party balance regime to be satisfied, the climate contribution must be kept between the lower and upper bounds of the optimized tax policy of both parties, as shown in Fig. 5.3e. If the climate damages fall bellow the lower bound, the external influence will always be $I_E < 0$ and favour the Lukewarmers (Fig. 5.4c through 5.4h). On the other hand, if climate damages surpass the upper tax bound it results in $I_E > 0$ favouring the Greens. Phase transitions from a paramagnetic state to a ferromagnetic one are not necessarily guaranteed, but these will most certainly occur if I_E crosses over the network's critical point. This is shown to always happen in favour of the Green party at later years.

Despite the behaviour of the coupling equation under these conditions being acceptable, it can certainly be improved. A dimensionless normalization of the coupling equation lacks a good basis for how to properly form a link between the DICE model and the network, in such a way that it accounts for the more intricate nuances of how the economic and climate modules of DICE interact. The evolution of the climate and tax terms behave in a rather nice manner and ultimately are only compared with the initial values, due to our dimensionless approach. We therefore implement a different normalization for the coupling equation, that preserves the units of different quantities of DICE. Our normalization proposal takes into account how each individual would perceive the effects of both climate and the tax terms. Society would perceive the effect of the tax term as the levied carbon tax imposed by policy makers (the marginal abatement in our simple case), multiplied by total yearly emissions. The climate damages term would be unchanged, only being adjusted according to the awareness parameter $\bar{\alpha}$. Both terms would be divided by the world population at the time, so that I_E translates into a contribution per individual. This normalized form of the coupling equation is now

$$\bar{I}_E(t) = L(t)^{-1} \left[\bar{\alpha} D(t) - \bar{\mu} E(t)\right] + \bar{\omega} F(\hat{T}) \tag{5.6}$$

where $L(t)$ is the world population at a given year, $\bar{\alpha}$ is the awareness parameter, $D(t)$ the yearly climate damages, $\bar{\mu}$ is the levied tax (the marginal abatement cost or social cost of carbon), $E(t)$ the total yearly carbon emissions and finally, the last term, account for future worries, which we will discuss later. We can see how this impacts the evolution of both the climate and tax terms in Fig. 5.5. The immediate consequence is the shape of the term curves, which are now no longer directly conditioned by the upper and lower bounds of the tax policies (even though the bounds still apply). This is because of our new choice of normalization, which now accounts

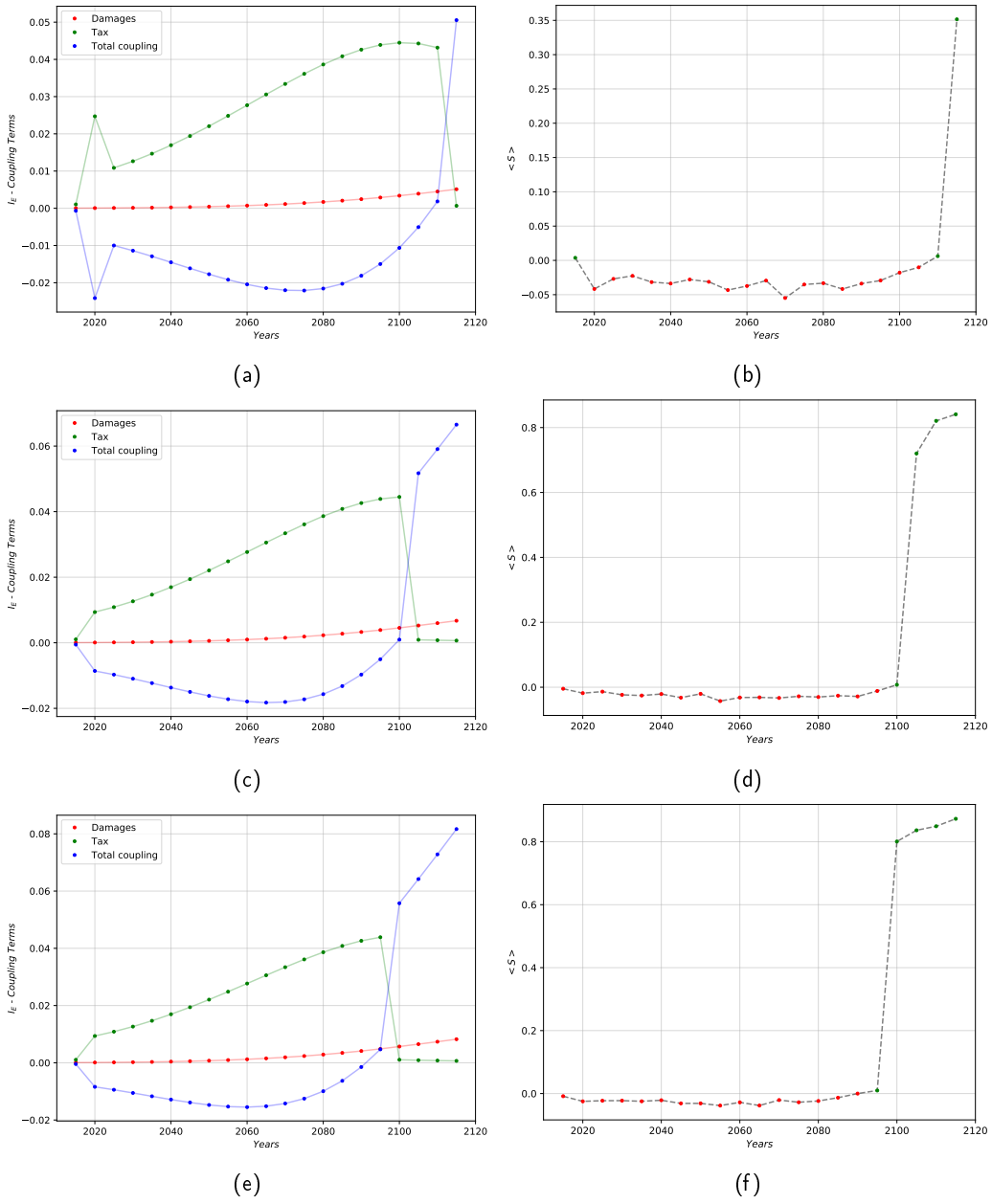
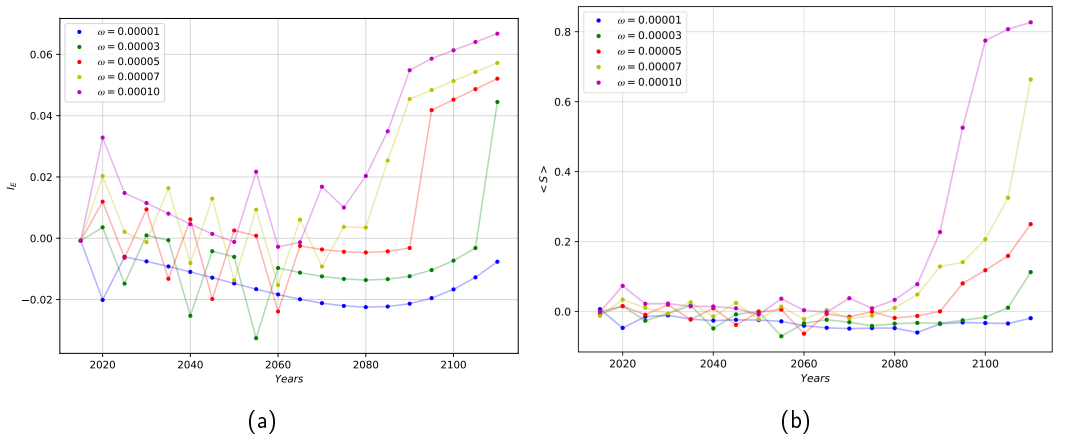


FIGURE 5.5: Effect of awareness $\bar{\alpha}$ on a normalization of the coupling equation that complies with DICE's units. The scale parameter is set at $\zeta = 0.0005$. On the left, the resulting external influence parameter I_E and its respective tax and damage contributions. On the right, the resulting opinion average. (a)(b) $\bar{\alpha} = 1.0$ (c)(d) $\bar{\alpha} = 1.5$ (e)(f) $\bar{\alpha} = 2.0$.



(a) (b)

FIGURE 5.6: Effect of the worry parameter $\bar{\omega}$ on the coupling equation. The scale parameter is set at $\zeta = 100$ and the awareness is $\bar{\alpha} = 100000$. On the left, the resulting external influence parameter I_E. On the right, the resulting opinion average $\langle S \rangle$.

for diminishing returns in the public perception of the tax in a more realistic manner. We observe a more stable party regime (in the examples shown, the dominance of the Lukerwarmers is evident - but this occurs for this particular set of chosen parameters and because we are yet to introduce a future worry term). At later years, a critical transition towards the Green party always occurs, as expected. A few last remarks about this choice of normalization: using the carbon tax to measure people's dislike of policy is somewhat arbitrary. Much of the tax would in real life be paid by companies, not by people, though part of the tax would indirectly affect people by making their products more expensive. Producers who do manage to abate more cheaply will not have to increase the price of their product by the carbon tax, but by its (lower) abatement cost. Finally, in real life, a tax does not simply vanish, but generates government income that can be used for other purposes such as higher unemployment benefits, further investments and other beneficial programmes. Our voters are obviously a very watered down representation and therefore shriek in disagreement about the tax, whereas in reality the economic "loss" due to abatement is only the abatement cost. In reality, when abatement increases, people pay less and less carbon tax because they don't emit as much (hence the importance of normalizing the tax term with the yearly emissions), but they still pay abatement cost - this effect might be something to worry about with respect to how much our coupling equations holds true.

5.3.3 Peak Temperature and Worry Parameter

A major difficulty of the coupling equation is including a term that is capable of conveying a society's worry about future events. As the consequences of climate change are long term, we don't expect immediate events to heavily influence election results. From the form of Eq. (5.1), we have seen that the carbon tax is the dominant short term effect on the coupling equation, bringing I_E towards negative values that favour the Lukewarmers. As climate damages ramp up, this will translate into a phase transition favouring the Greens at later years. However, by then the climate damages are considerably high and the average world temperature also. Early abatement is critical in order to mitigate the rampant increase of the average world temperature, but without a term that makes our population worry about the future impacts of climate change, we will never see the implementation of policies

by the Green party. We have previously introduced this worry term and how we construct it based on the peak temperature anomaly projected by each party, but we will now show the actual impact of its implementation.

From this point onwards, we will always take our coupling equation (5.1) to be of the form

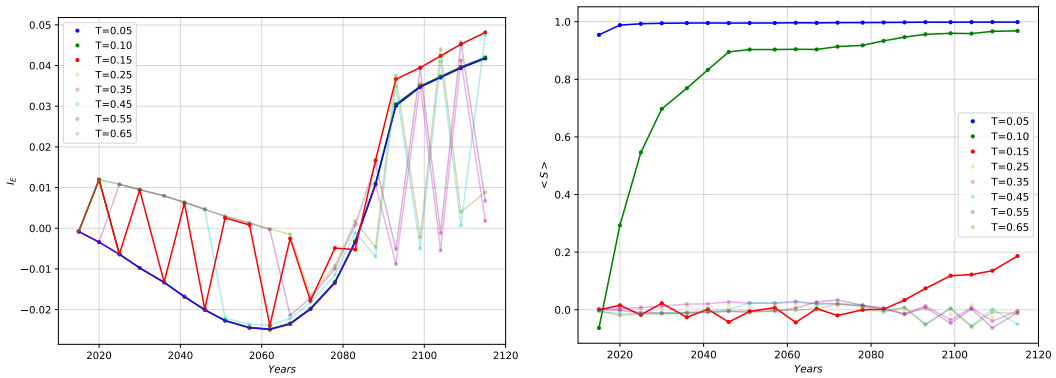
$$I_E(t) = \zeta\{\frac{1}{L(t)}\left[\bar{\alpha}D(t) - \bar{\mu}E(t)\right] + \bar{\omega}F(\hat{T})\} \tag{5.7}$$

with $F(\hat{T})$ as the identity, such that it reduces to the form of Eq. (5.2). The value of parameters change considerably from previous examples due to our choice of normalization, but this is a recurring issue with our model - parameters have to be specific to the chosen normalization and the network properties.

We chose a scale parameter $\zeta = 100$ and awareness of $\bar{\alpha} = 100000$ for our subsequent examples. Fig. 5.6 presents the results for a range of different values of the worry parameter $\bar{\omega}$ where the network stays in a paramagnetic regime where occasional political party shifts occur. We can see that for low values of $\bar{\omega}$, the coupling equation is for the most part negative, locking the system into a permanent bias towards the Lukewarmers. A more interesting behaviour is observed when we increase the worry parameter, which leads to phase transitions towards a ferromagnetic state in favour of the Greens. Like we have seen before, this is to be expected at later years due to how the climate damage term and tax term evolve over time. The affect of the worry term is to allow for party shifts to occur when the system is still relatively paramagnetic, so that the ruling party alternates over the years. This is more prominent for the regime where $\bar{\omega} = 0.00005$, which is the closest regime to displaying a somewhat realistic political party balance behaviour in the first century. For the regime where $\bar{\omega} = 0.00007$, the system exhibits the same balanced behaviour until it transitions into a ferromagnetic phase at year 2115. On the upper end, when we take $\bar{\omega} = 0.00010$ the phase transition happens at year 2095, but we point out that despite the system being in a paramagnetic state in the previous years, the party that stays in power for the most part is the Greens - this is a result of I_E being just high enough to nudge the system with a slight, but noticeable bias towards the Green party.

5.3.4 Network Temperature

Previously, we discussed how the temperature of the network itself can be interpreted as a measure of social unrest of a given population. One could port this analogy to a society that is undergoing problems related with crime, inequality, racial discrimination and even disagreements regarding climate change and economic turmoil. For this reason, it is worth seeing how this affects the behaviour of our model. The results are shown in Fig. 5.7, where we highlight the most interesting regimes. Throughout all of our simulations, we have always set the network temperature at $T = 0.15$, which is the standard network temperature in the literature [33]. This regime is highlighted and plotted in red, and it exhibits the same behaviour seen in Fig. 5.6 when $\bar{\omega} = 0.00005$). However, when we lower the network temperature the system very quickly settles in a ferromagnetic phase (see the blue and green curves in Fig. 5.7b). When the network temperature is increased, the system stays locked in a paramagnetic phase. This behaviour is in agreement with the properties of the network and how its temperature impacts the effect that the external influence I_E has on swaying the opinions of individuals (see Fig. 3.13 and 3.14). This reinforces the fact that societies need to be in a very specific condition in order for



(a) (b)

FIGURE 5.7: The behaviour of the (a) external influence parameter I_E and (b) opinion average $\langle S \rangle$; for networks at different network temperature T. The scale parameter is set at $\zeta = 100$, the awareness is $\bar{\alpha} = 100000$, and the worry parameter is $\bar{\omega} = 0.00005$. Highlighted are the lower network temperature regimes where phase transitions occur. For high network temperature, the system is locked in a paramagnetic state.

public opinion to be shaped by external factors that are, in our model, connected to climate and economic events. Societies that have a low network temperature are conformists, where the lack of societal unrest and disagreements amongst its population will lead to everyone voting for the same party. Higher network temperature societies on the other hand are riddled with so much unrest that it's much too divided in order for one particular political party to take the upper hand. Here we give a nod to the highly politically polarized state of the United States of America.

An interesting thing that the network temperature tests clearly showcases is the year where climate related concerns overtake tax concerns. In Fig. 5.7a we can see that the I_E curve takes an upward turn at year 2090. The irregularity between distinct years is a result of the tax term associated with each party, but these are nonetheless within an upper and lower bound. These upper and lower bounds of I_E are once again directly linked to the parties' climate abatement policies as seen in Fig. 5.3e.

As remark, it is not immediately clear how to regulate the network temperature in such a way that it matches the world's current state of affairs or even how to implement a mechanism that allows for a dynamic network temperature that is linked to the time evolution of the real-DICE model.

5.3.5 Authority

Finally, we are interested in exploring how the distribution of authority A_{ij} impacts our model. According to Grabowski and Kosiński, the authority can be interpreted as a measure of a population's education. We instead argue that it more accurately represents an individual's resistance to changing their opinion - a measure of stubbornness, if you will. In order for authority to be somewhat comparable to one's level of education (we don't necessarily mean the level of one's academic achievements, but how well informed an individual is on a given subject) we have to ensure that the distribution of A_{ij} makes somewhat sense with the randomly allocated initial values for S_{ij}. It is possible to specifically assign a certain S_{ij} to individuals that have high authority on the chosen subject. In our case, if we assume that high authority relates to how well educated an individual is on the topic of man-made

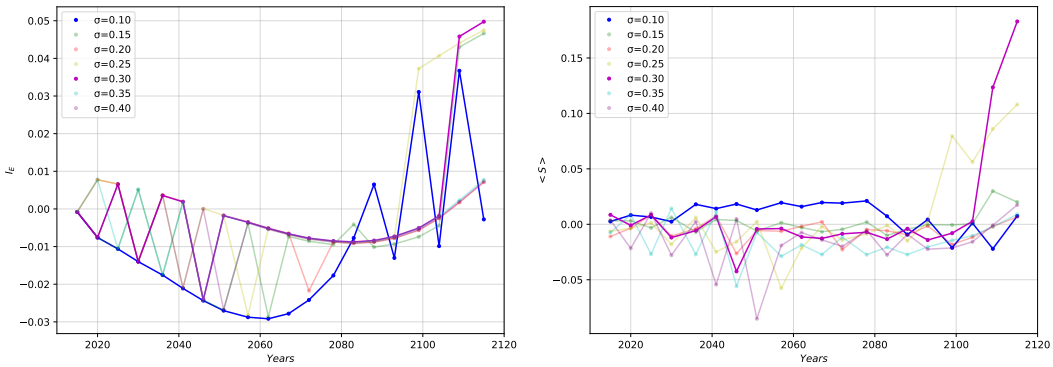


FIGURE 5.8: The behaviour of the (a) external influence parameter I_E and (b) opinion average $\langle S \rangle$; for networks with different variance σ, i.e. different distributions of authority A_{ij}. The scale parameter is set at $\zeta = 100$, the awareness is $\bar{\alpha} = 100000$, and the worry parameter is $\bar{\omega} = 0.00004$. Highlighted are the two regimes discussed in the text.

climate change, we can then assign $S_{ij} = 1$ to individuals with high authority. However, it is clear that that the system will very quickly reach $\langle S \rangle = 1$ under these conditions. As this behaviour is unrealistic, it cannot be accurately linked to a population's education. The tests performed for different distributions of authority (done by taking different values of the variance σ) are shown in Fig. 5.8. These results (see Fig. 5.8b) are quite inconclusive, as there is no apparent trend in the opinion average $\langle S \rangle$. The system seems to be in a paramagnetic phase in most of the plotted curves, with the regimes $\sigma \approx 0.30$ exhibiting what looks like a transition into a paramagnetic phase at later years (note that $\sigma = 0.30$ is the authority distribution we've taken throughout all our simulations). Higher σ seems to indicate that the effect of the external influence I_E into swaying opinion is diminished. This is on a par with our interpretation of A_{ij} as a measure of stubbornness, since individuals with higher A_{ij} are not only more likely to be able to enforce their opinion on others, but also they are less likely to have their own opinions change. A strange result is the case when $\sigma = 0.10$: we can see that when $I_E < 0$, we have $\langle S \rangle > 0$ and when $I_E > 0$, we get $\langle S \rangle < 0$. This seems to suggest that for low distributions of authority the resulting opinion average is opposite to the external influence I_E. An explanation for this is currently lacking. We do not exclude the possibility that more interesting regimes for $\sigma \neq 0.3$ exist. If they do, however, an exhaustive search that would require a completely different set of coupling parameters (ζ; $\bar{\alpha}$; and $\bar{\omega}$) would be needed. Due to time constraints we did not further explore this. For this reason, future work would be needed into potential affects of authority and its interpretation in the context of our coupled-DICE model.

5.4 Atmospheric Temperature Homeostasis

Ultimately, our intent is to show how the electoral outcomes of democratic societies can have an impact on the average world temperature, and under which conditions we can observe a regulatory mechanism of the Earth's temperature based on human activity. We know, of course, that economic activity leads to an increase in the average world temperature through carbon emissions. By adding abatement policies we can lower our collective emissions in the future and consequentially lessen the impact of anthropogenic activity on the climate system.

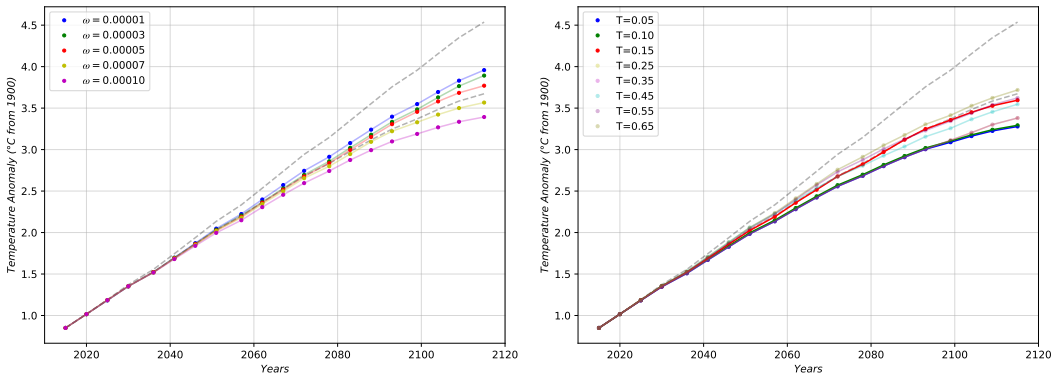


FIGURE 5.9: The average atmospheric temperature anomaly for the different regimes we've explored. (a) Worry parameter ($\bar{\omega}$) regimes. (b) Network temperature (T) regimes.

We can see how the different regimes for the worry parameter $\bar{\omega}$ and network temperature T lead to different curves of the atmospheric temperature anomaly and how it compares to established scenarios (Fig. 5.9). In both plots the dotted grey lines correspond to the DICE-2016 baseline (upper) and optimal policy (lower) scenarios. Since our coupled-DICE is itself bound by the optimal policies of both parties, the outcomes of the different regimes are always within this range and this ports over to the average world temperature. Effective tackling of climate change and the rampant increases in world temperature that follow ultimately require active and ambitious policy to be implemented. Since we have assumed that both the Lukewarmers and the Greens will do this (despite having different views on how to do so), the world temperature curves will always be lower than the baseline, "business as usual" scenario. The most ambitious policies are obviously the ones of the Green party. Our model sets the lower bound for world temperature changes to the curve where the system is always ferromagnetic in favour of the Greens ($\langle S \rangle > 0$) throughout the entire simulation. These are the $\bar{\omega} = 0.00010$ purple curve in Fig. 5.9a and $T = 0.05$ blue curve in Fig. 5.9b. The upper bound happens when the Lukewarmers are always in power and corresponds to the $\bar{\omega} = 0.00001$ blue curve in Fig. 5.9a. This upper bound never happens in the network temperature regimes we have studied (Fig. 5.9b), but this is because of how the system is locked in a paramagnetic phase and subject to fluctuations for high network temperature. Under our political party balance regimes, chosen for the sake of realism, the resulting curves are always intermediate. Therefore, for a two party system, we can easily conclude that the regulation of global warming is immediately bound by the climate ideologies of the parties that are up for election; which in our examples are set by a simple distinction in the climate sensitivity parameter λ of the DICE-2016 policy optimization.

Chapter 6

Conclusion

Throughout this thesis, we have taken steps in the exploration of the complexity that arises from the interconnectedness of fields as distinct as physics, economics and sociology and its interesting results. We have developed a framework for how democratic societies play a major role in the future development of the climate change crisis that humanity is facing. The coupled DICE model that we have devised provides a simple way to explore the impact of elections on the climate systems and the world's economy. Through an analysis of multiple regimes, we have seen how human perception of the consequences of climate change affects opinion formation and electoral outcomes. The role of taxation for purposes of tackling the climate emergency is clearly the main concern of voters, pushing them to prefer political parties that enforce lower abatement policies. It is only when voters care about the immediate climate damages and future impacts that we see pro-climate parties winning elections and enforcing their higher abatement policies. Transitions of voter behaviour from a relatively stable balance of power between parties are also observed at later years in favour of pro-climate parties, but the consequences of climate change may well be too severe at that stage for us effectively do something about it. Ultimately, we see how these different regimes lead to atmospheric temperature anomaly scenarios that are within a given range, set by the party ideology regarding climate change. If this can be considered to be a homeostatic regime, it leaves much to be desired. When constrained by the policies of the Greens and the Lukewarmers, we don't see how climate change can be "fixed" in a way that won't lead to significantly high atmospheric temperature anomalies. It seems to suggest that tackling the challenges of climate change (even under optimal policy regimes) won't be enough to prevent the consequences of rampant anthropogenic activity since the industrial revolution. We also do not account for the nuances of dynamic networks and how the network properties could be preserved under such circumstances, or even how opinion formation would operate with more than two political parties. Further development on this subject could also help shed light in the role that the authority of individuals plays and how we can make it more dynamic, as well as taking into account shifting political ideologies and learning behaviours for both parties. In the end, it will seem that fixing climate change will require decisive coordinated global action of unprecedented ambition.

Appendix A

Network Generator Algorithm

Here, we present a simple version of the network generator algorithm.

```
import numpy as np
import networkx as nx
import matplotlib.pyplot as plt
from matplotlib.font_manager import FontProperties

font = FontProperties()
font.set_style('italic')

from scipy.stats import truncnorm
from mpl_toolkits.axes_grid1 import make_axes_locatable

from datetime import datetime
startTime = datetime.now()

# MAIN MODEL PARAMETERS
L = 100              # size of the square grid
N = L**2             # number of individuals/nodes (L*L)
Lg = 20              # size of the local groups
Ng = Lg**2           # number of individuals/nodes per local group
Lgx = int(L/Lg)      # required for local group identifier

gamma = 3.0          # exponent of the power law distribution for the
    connectivity k
kmin = 10            # minimum number of connections of a node
kmax = 100           # maximum number of connections of a node
pc = 0.5             # probability to create second level connections

it = 2500            # number of iterations for the NETWORK generator

# acceptance counters

count_it = np.linspace(1, it, it, dtype = np.int32)

saturation = np.zeros(it)
avg_clustering = np.zeros(it)

count_primary = np.zeros(it)
count_secondary = np.zeros(it)
count_rejected = np.zeros(it)


# SECONDARY MODEL PARAMETERS

variance = 0.3  # sigma**2 (needed for truncated Gaussian distribution)

a = Lg      # parameter for distance probability function
b = Lg/4    # parameter for distance probability function
```

```
lcrit = 100  # critical distance parameter for node selection (improves
    speed)

# ------------------------------- #
# ---------- MAIN BODY ---------- #
# ------------------------------- #

# MATRIX GENERATORS

Sij = 2*np.random.randint(2, size=(L,L))-1     # generates the matrix Sij

# algorithm that ensures opinion average is 0 at the start (only works for
     even N)!
while np.sum(Sij)/N != 0:

    if (N % 2) != 0:
        break

    if np.sum(Sij)/N > 0:

        l1 = np.random.randint(0, L)
        l2 = np.random.randint(0, L)

        if Sij[l1,l2] > 0:

            Sij[l1,l2] = (-1)*Sij[l1,l2]

        else : continue

    else:

        l1 = np.random.randint(0, L)
        l2 = np.random.randint(0, L)

        if Sij[l1,l2] < 0:

            Sij[l1,l2] = (-1)*Sij[l1,l2]

        else: continue

print(f'Opinion␣average={np.sum(S)/N}')

Kij = np.zeros(shape=(L,L))    #generator matrix for kij
count = -1                     #counting index

# connectivity matrix generated through a power law distribution
# re-maping function
def power_law(kmin, kmax, y, gamma):
    return ((kmax**(-gamma+1)-kmin**(-gamma+1))*y+kmin**(-gamma+1.0))
        **(1.0/(-gamma+1.0))

scale_free_dist = np.zeros(N, float)     # empty distribution matrix

# transforming a uniform distribution to a truncated power law shape
for n in range(N):
    scale_free_dist[n] = int(power_law(kmin, kmax,np.random.uniform(0,1),
        gamma))

# assignment of the distribution values to Kij matrix
for i in range(L):
    for j in range(L):
        count +=1
```

```
        Kij[i,j] = scale_free_dist[count]

# checking for the probability distribution function
kcount = np.zeros(101)

for n in range(N):
    ki = int(scale_free_dist[n])
    kcount[ki]+=1

# Gaussian distribution to generate authority matrix
mu, sigma = 0, np.sqrt(variance) # mean and standard deviation

#re-parametrized truncated normal distribution function
def get_truncated_normal(mean=0,sd=1,low=0,upp=10):
    return truncnorm(
            (low-mean)/sd,(upp-mean)/sd,loc=mean,scale=sd)

X = get_truncated_normal(mean=0,sd=sigma,low=0,upp=1)

A = X.rvs(N)     #authority values from distribution

Aij = np.zeros(shape=(L,L)) #authority matrix
count = -1  #counting index

#assignment of values to authority matrix
for i in range(L):
    for j in range(L):
        count +=1
        Aij[i,j] = A[count]

# saving the relevant arrays [optional]
#np.save("S [L_%d;gamma_%d;pc_%f;LG_%d].npy" % (L,gamma,pc,Lg), Sij)
#np.save("K [L_%d;gamma_%d;pc_%f;LG_%d].npy" % (L,gamma,pc,Lg), Kij)
#np.save("A [L_%d;gamma_%d;pc_%f;LG_%d].npy" % (L,gamma,pc,Lg), Aij)

# (all matrix generator plots at the end [optional])

# NETWORK GENERATOR (slowest part)

kij = np.zeros(shape=(L,L)) # real number of connections of each node/
    individual
C = np.zeros(shape=(L,L))   # clusterring coefficient matrix

# generates a plus or minus sign with equal probability
def sign():
    return 1 if np.random.random()< 0.5 else -1

# creates a grid graph
G = nx.grid_2d_graph(L,L)

# position the nodes with equal distance in a lattice structure
pos = dict( (n, n) for n in G.nodes() )

# deletes all edges from the graph G, leaving only the nodes
G = nx.create_empty_copy(G)

# plots the empty graph (just to check) [optional]
#figG = plt.figure(figsize=(8,8))
#nx.draw(G,pos=pos,node_size=1,node_color='k',width=0.5)
#plt.axis('off')
#plt.savefig('empty_grid.pdf', format='pdf', bbox_inches='tight')
#plt.show()
```

```
# algorithm that generates second level connections
def second_order(i,j,n,m):

    # secondary connections (i,j) --> (n,m)
    neigh = list(G.neighbors((n,m)))            # list of neighbours of
        node (n,m)
    for z in range(len(neigh)):                 # iterate over all
        neighbours

        if (i,j) == neigh[z]:                   # ignores self: (i,j)
            continue

        if G.number_of_edges((i,j),neigh[z]) < 1:

            roll = np.random.random()

            if roll < pc:

                #index selection
                tupz = neigh[z]
                z1 = tupz[0]
                z2 = tupz[1]

                if kij[i,j] > kij[z1,z2]:
                    count_rejected[tt] += 1
                    continue

                if kij[i,j] < Kij[i,j] and kij[z1,z2] < Kij[z1,z2]:
                    G.add_edge((i,j),(z1,z2))   # adds a second level
                        connection!
                    kij[i,j]+=1
                    kij[z1,z2]+=1
                    count_secondary[tt] += 1
                else:
                    count_rejected[tt] += 1
                    continue



# algorithm for creating interpersonal connections
def Interpersonal():
    for i in range(L):
        for j in range(L):

            if kij[i,j] == Kij[i,j]:  # checks if more connections are
                allowed
                count_rejected[tt] += 1
                continue

            # selects the random variables for indice pairing
            l1 = np.random.randint(0, lcrit)
            l2 = np.random.randint(0, lcrit)

            # selects the indices for the pairing
            n = i+sign()*l1
            m = j+sign()*l2

            # ensures that the index value is never out of the range [0,L]
            if n > L-1:
                n = n-L
            if m > L-1:
                m = m-L
            if n < 0:
```

```
            n = n+L
        if m < 0:
            m = m+L

        if i == n and j == m:    # in case it selects the same node,
            skip
            continue

        # distance between individuals
        l = np.sqrt((i-n)**2+(j-m)**2)

        # probability to form a connection between individuals P(l)
        P = 1/(1+np.exp((l-a)/b))+0.001*((L-l)/L)

        # probability "dice roll"
        Proll = np.random.random()

        # connection generator algorithm now goes through all the
            conditional checks

        # first level connection between (i,j) and (n,m)
        if Proll < P:
            if G.number_of_edges((i,j),(n,m)) < 1:
                if kij[i,j] > kij[n,m]:
                    count_rejected[tt] += 1
                    continue
                if kij[i,j] < Kij[i,j] and kij[n,m] < Kij[n,m]:
                    G.add_edge((i,j),(n,m))    # adds a first level
                        connection!
                    kij[i,j]+=1
                    kij[n,m]+=1
                    count_primary[tt] += 1
                else:
                    count_rejected[tt] += 1
                    continue

        # second level connections algorithm
        if G.number_of_edges((i,j),(n,m)) == 1:

            # secondary connections (n,m) --> (i,j)
            second_order(n,m,i,j)

            # secondary connections (i,j) --> (n,m)
            second_order(i,j,n,m)


# runs the network generator it times
for tt in range(it):
    Interpersonal()

    # calculates saturation and clustering for each iteration [optional]
    # (computationaly expensive!)
    #saturation[tt] = np.sum(kij)/np.sum(Kij)
    #avg_clustering[tt] = nx.average_clustering(G)


print(f'Computation time = {datetime.now() - startTime}')

grad = Kij - kij      # visual check of the "quality" of the network

print(f'Network saturation = {np.sum(kij)/np.sum(Kij)}')     # network
    saturation
```

```
print(f'Network clustering = {nx.average_clustering(G)}')    # network
    clustering

count_accepted = count_primary + count_secondary

## GENERATOR PLOTS [OPTIONAL] ##
# color map of Sij matrix (1,-1)
fig, ax = plt.subplots(figsize=(8,8))
plt.title('Color map of Sij')
plt.xticks([])
plt.yticks([])
im = plt.imshow(Sij, cmap='coolwarm')

divider = make_axes_locatable(ax)
cax = divider.new_vertical(size="5%", pad=0.5, pack_start=True)
fig.add_axes(cax)
fig.colorbar(im, cax=cax, orientation="horizontal")

plt.savefig('colormap_Sij_init.pdf', format='pdf', bbox_inches='tight')
plt.show()
plt.close()

# color map of Kij generator matrix (kmin,kmax)
fig, ax = plt.subplots(figsize=(8,8))
plt.title('Color map of Kij')
plt.xticks([])
plt.yticks([])
im = plt.imshow(Kij, cmap='YlOrBr')

divider = make_axes_locatable(ax)
cax = divider.new_vertical(size="5%", pad=0.5, pack_start=True)
fig.add_axes(cax)
fig.colorbar(im, cax=cax, orientation="horizontal")

plt.savefig('colormap_Kij_gen.pdf', format='pdf', bbox_inches='tight')
plt.show()
plt.close()

# color map of Aij  matrix (0,1)
fig, ax = plt.subplots(figsize=(8,8))
plt.title('Color map of Aij')
plt.xticks([])
plt.yticks([])
im = plt.imshow(Aij, cmap='YlGnBu')

divider = make_axes_locatable(ax)
cax = divider.new_vertical(size="5%", pad=0.5, pack_start=True)
fig.add_axes(cax)
fig.colorbar(im, cax=cax, orientation="horizontal")

plt.savefig('colormap_Aij.pdf', format='pdf', bbox_inches='tight')
plt.show()
plt.close()


#Power law distribution
kx = np.round(np.linspace(kmin,kmax,kmax-kmin))

plt.figure(figsize=(7,5))
plt.title('Distribution of connectivity Kij')
plt.xlabel('k')
plt.ylabel('P(k)')
plt.yscale('log')
```

```
plt.xscale('log')
plt.plot(kx,kcount[kmin:kmax]/N,'r*')

plt.savefig('Kij_dist.pdf', format='pdf', bbox_inches='tight')
plt.show()
plt.close()

xx = np.linspace(0,1,100)
plt.figure(figsize=(7,5))
plt.title('Distribution␣of␣authority␣Aij')
plt.xlabel('A')
plt.ylabel('P(A)')
plt.hist(A, 20, density=True)
plt.plot(xx, 2/(sigma * np.sqrt(2 * np.pi)) *
                np.exp( - (xx - mu)**2 / (2 * sigma**2) ),
         linewidth=1, color='r')
plt.show()
plt.close()

#Color map of Saturation
fig, ax = plt.subplots(figsize=(8,8))
plt.title('Saturation␣Kij-kij')
plt.xticks([])
plt.yticks([])
im = plt.imshow(grad)

divider = make_axes_locatable(ax)
cax = divider.new_vertical(size="5%", pad=0.5, pack_start=True)
fig.add_axes(cax)
fig.colorbar(im, cax=cax, orientation="horizontal")

plt.savefig('network_saturation.pdf', format='pdf', bbox_inches='tight')
plt.show()
plt.close()

plt.figure(figsize=(8,6))
plt.title('Acceptance')
plt.xlabel('Generator␣iterations',fontproperties=font)
plt.ylabel('Counts',fontproperties=font)
plt.yscale('log')
plt.xscale('log')
plt.grid(alpha=0.5)
plt.plot(count_it,count_rejected,'r-',label='Rejected')
plt.plot(count_it,count_accepted,'g-',label='Accepted␣(Total)')
plt.plot(count_it,count_primary,'c-',label='Accepted␣(Primary)')
plt.plot(count_it,count_secondary,'b-',label='Accepted␣(Secondary)')
plt.legend()

plt.savefig('acceptance.pdf', format='pdf', bbox_inches='tight')

plt.figure(figsize=(8,6))
plt.title('Saturation')
plt.xlabel('Generator␣iterations',fontproperties=font)
plt.ylabel('Saturation',fontproperties=font)
plt.grid(alpha=0.5)
plt.plot(count_it,saturation,'b-')

#np.save("count_it.npy", count_it)
#np.save("count_accepted.npy", count_accepted)
#np.save("count_rejected.npy", count_rejected)
#np.save("count_primary.npy", count_primary)
#np.save("count_secondary.npy", count_secondary)
```

Appendix B

Opinion Dynamics Algorithm

Here, we present a simple version of the Ising-like opinion mechanism.

```
import random
from pack_unpack import unpack_matrix

T = 0.15              # network temperature
Iex = 0               # external stimulation parameter
steps = 1000000       # simulation steps

print(f'Initial opinion average = {np.sum(Sij)/N}')

#local field calculations

#breaking the Aij matrix into local groups
#(breaking Sij needs to be done on the go, because it's constantly updated
    !)
uA = unpack_matrix(Aij,Lgx,Lgx)

Sp = np.zeros(steps)

for tt in range(steps):

    i = random.randrange(0,L)
    j = random.randrange(0,L)

    hij = 0
    iisum = 0
    losum = 0

    uS = unpack_matrix(Sij,Lgx,Lgx) #local groups of Sij

    neigh1 = list(G.neighbors((i,j)))
    for z in range(len(neigh1)):

        #index selection
        tup1 = neigh1[z]
        z1 = tup1[0]
        z2 = tup1[1]

        #contribution of interpersonal connections
        iisum += Aij[z1,z2]*Sij[z1,z2]

    #local group identification
    ix = np.floor(i/Lg)
    jx = np.floor(j/Lg)

    #contribution of local group
    losum = np.sum(uS[(ix, jx)]*uA[(ix, jx)])
```

```
        hij = (1/kij[i,j])*(iisum+(losum/Ng))+Iex

        #opinion switch mechanism
        if T == 0:

            if hij*Sij[i,j] >= 0:

                pij = 0

            else:

                pij = (1-Aij[i,j])

                #probability "dice roll"
                Proll = np.random.random()

                if Proll <= pij:

                    Sij[i,j] = (-1)*Sij[i,j]

        else:

            if hij*Sij[i,j] > 0:

                pij = (1-Aij[i,j])*np.exp((-hij*Sij[i,j])/T)

                #probability "dice roll"
                Proll = np.random.random()

                if Proll <= pij:

                    Sij[i,j] = (-1)*Sij[i,j]

            else:

                pij = (1-Aij[i,j])*(1-np.exp((hij*Sij[i,j])/T))

                #probability "dice roll"
                Proll = np.random.random()

                if Proll <= pij:

                    Sij[i,j] = (-1)*Sij[i,j]


        Sp[tt] = np.sum(Sij)/N

        #print(f'Step = {tt} ; <S> = {np.sum(Sij)/N}')

        if tt % 10000 == 0:
            plt.figure(figsize=(7,5))
            plt.imshow(Sij, cmap='coolwarm')
            plt.axis('off')
            plt.savefig('frame_%d.png' % tt, format='png')
            plt.close()

print(f'Final opinion average={np.sum(Sij)/N}')


# ---------------------------- auxiliary functions
    ------------------------------ #

def unpack_matrix(m, num_of_rows, num_of_columns):
```

```
    # sanity checks
    n_rows, n_cols = m.shape
    if (n_rows % num_of_rows != 0 or n_cols % num_of_columns != 0):
        print(f'cannot␣unpack␣{m.shape}␣with␣dimensions␣({num_of_rows},\
␣␣␣␣␣␣␣␣␣␣␣␣{num_of_columns})␣without␣losing␣information')

    # calculate sub-matrices sizes
    sub_matrix_width = n_cols // num_of_columns
    sub_matrix_height = n_rows // num_of_rows

    unpacked = {}
    for i in range(n_rows // sub_matrix_height):
        for j in range(n_cols // sub_matrix_width):
            unpacked[(i, j)] = m[sub_matrix_height*i:sub_matrix_height*(i
                +1),
            sub_matrix_width*j:sub_matrix_width*(j+1)]

    return unpacked

def pack_matrix(upm, num_of_rows, num_of_cols):
    # reconstruct original shape
    sub_matrix_width, sub_matrix_height = upm[(0, 0)].shape
    full_matrix_width, full_matrix_height = sub_matrix_width*num_of_rows,
    sub_matrix_height*num_of_cols
    full_matrix = np.zeros((full_matrix_width, full_matrix_height))

    # copy submatrices inside the full matrix at the right locations
    for i in range(num_of_rows):
        for j in range(num_of_cols):
            full_matrix[sub_matrix_height*i:sub_matrix_height*(i+1),
            sub_matrix_width*j:sub_matrix_width*(j+1)] = upm[(j, i)]

    return full_matrix
```

Appendix C

Coupled-DICE Algorithm

Finally, our implementation of DICE-2016 in Python with the coupling equation and network dynamics. (The optimization function for DICE is built in a similar way and makes use of a Sequential Least Squares Programming (SLSQP) minimization function.)

```
import numpy as np
from numba import njit
from numba import guvectorize
from numba import float64

from matplotlib import pyplot as plt
from pack_unpack import unpack_matrix

from DICE_optimizer import DICE_optimizer

from datetime import datetime
startTime = datetime.now()

#** Set **
t = np.arange(1, 22)
NT = len(t)

#** Parameters **

fosslim = 6000    # Maximum cumulative extraction fossil fuels (GtC); CCum
tstep   = 5       # Years per Period
ifopt   = 0       # Indicator where optimized is 1 and base is 0

#** Preferences **

alpha = 1.45     # Elasticity of marginal utility of consumption
rho = 0.015      # Initial rate of social time preference per year

#** Population and technology **

gamma = 0.300       # Capital elasticity in production function
    /.300/
pop0 = 7403         # Initial world population 2015 (millions)
    /7403/
popadj = 0.134      # Growth rate to calibrate to 2050 pop projection
    /0.134/
popasym = 11500     # Asymptotic population (millions)
    /11500/
delk   = 0.100      # Depreciation rate on capital (per year)
    /.100/
q0 = 105.5          # Initial world gross output 2015 (trill 2010 USD)
    /105.5/
```

```
k0 = 223            # Initial capital value 2015 (trill 2010 USD)
    /223/
a0 = 5.115          # Initial level of total factor productivity
    /5.115/
ga0 = 0.076         # Initial growth rate for TFP per 5 years
    /0.076/
dela = 0.005        # Decline rate of TFP per 5 years
    /0.005/

#** Emissions parameters **

gsigma1 = -0.0152  # Initial growth of sigma (per year)
    /-0.0152/
delsig = -0.001     # Decline rate of decarbonization (per period)
    /-0.001/
eland0 = 2.6        # Carbon emissions from land 2015 (GtCO2 per year)
    /2.6/
deland = 0.115      # Decline rate of land emissions (per period)
    /.115/
eind0 = 35.85       # Industrial emissions 2015 (GtCO2 per year)
    /35.85/
miu0 = 0.03         # Initial emissions control rate for base case 2015
    /.03/

#** Carbon cycle **
#* Initial Conditions *

mat0 = 851       # Initial Concentration in atmosphere 2015 (GtC)
    /851/
mup0 = 460       # Initial Concentration in upper strata 2015 (GtC)
    /460/
mlo0 = 1740      # Initial Concentration in lower strata 2015 (GtC)
    /1740/
mateq = 588      # mateq Equilibrium concentration atmosphere  (GtC)
    /588/
mupeq = 360      # mupeq Equilibrium concentration in upper strata (GtC)
    /360/
mloeq = 1720     # mloeq Equilibrium concentration in lower strata (GtC)
    /1720/

#* Flow parameters, denoted by Phi_ij in the model *
b12 = 0.12    # Carbon cycle transition matrix
    /.12/
b23 = 0.007   # Carbon cycle transition matrix
    /0.007/

#* These are for declaration and are defined later *
b11 = None    # Carbon cycle transition matrix
b21 = None    # Carbon cycle transition matrix
b22 = None    # Carbon cycle transition matrix
b32 = None    # Carbon cycle transition matrix
b33 = None    # Carbon cycle transition matrix
sig0 = None   # Carbon intensity 2010 (kgCO2 per output 2005 USD 2010)

#** Climate model parameters **

t2xco2 = 3.1        # Equilibrium temp impact (oC per doubling CO2)
    /3.1/
fex0 = 0.5          # 2015 forcings of non-CO2 GHG (Wm-2)
    /0.5/
fex1 = 1.0          # 2100 forcings of non-CO2 GHG (Wm-2)
    /1.0/
```

```
tocean0 = 0.0068    # Initial lower stratum temp change (C from 1900)
    /.0068/
tatm0 = 0.85        # Initial atmospheric temp change (C from 1900)
    /0.85/
c1 = 0.1005         # Climate equation coefficient for upper level
    /0.1005/
c3 = 0.088          # Transfer coefficient upper to lower stratum
    /0.088/
c4 = 0.025          # Transfer coefficient for lower level
    /0.025/
fco22x = 3.6813     # eta: Forcings of equilibrium CO2 doubling (Wm-2)
    /3.6813/

#** Climate damage parameters **

a10 = 0         # Initial damage intercept
                                        /0/
a20 = None      # Initial damage quadratic term
a1 = 0          # Damage intercept
                                                /0/
a2 = 0.00236    # Damage quadratic term
    /0.00236/
a3 = 2.00       # Damage exponent
    /2.00/

#** Abatement cost **

expcost2 = 2.6    # Theta2 in the model, Eq. 10 Exponent of control cost
    function    /2.6/
pback = 550       # Cost of backstop 2010$ per tCO2 2015
    /550/
gback = 0.025     # Initial cost decline backstop cost per period
    /.025/
limmiu = 1.2      # Upper limit on control rate after 2150
    /1.2/
tnopol = 45       # Period before which no emissions controls base
            /45/
cprice0 = 2       # Initial base carbon price (2010$ per tCO2)
             /2/
gcprice = 0.02    # Growth rate of base carbon price per year
    /.02/

#** Scaling and inessential parameters
#* Note that these are unnecessary for the calculations
#* They ensure that MU of first period's consumption =1 and PV cons = PV
    utilty
scale1 = 0.0302455265681763  # Multiplicative scaling coefficient
    /0.0302455265681763/
scale2 = -10993.704          # Additive scaling coefficient
    /-10993.704/;

#* Parameters for long-run consistency of carbon cycle
#(Question)
b11 = 1 - b12
b21 = b12*mateq/mupeq
b22 = 1 - b21 - b23
b32 = b23*mupeq/mloeq
b33 = 1 - b32

#* Further definitions of parameters
a20 = a2
sig0 = eind0/(q0*(1-miu0)) # derived from industrial emissions equation at
     t=0
```

```
lam = fco22x/ t2xco2 # equilibrium climate sensitivity (?)

#** Initialization **

# Labor/Population
l = np.zeros(NT)
l[0] = pop0

# Total Factor Productivity (TFP)
al = np.zeros(NT)
al[0] = a0

# Baseline carbon intensity growth
gsig = np.zeros(NT)
gsig[0] = gsigma1

# Baseline carbon intensity
sigma = np.zeros(NT)
sigma[0]= sig0

# TFP growth rate dynamics
ga = ga0 * np.exp(-dela*5*(t-1))

# Backstop price
pbacktime = pback * (1-gback)**(t-1)

# Emissions from deforestration
etree = eland0*(1-deland)**(t-1)

# Discount factor arising from pure time preference (R)
rr = 1/((1+rho)**(tstep*(t-1)))

# Parametrization of exogenous radiative forcing (Fex);
forcoth = np.full(NT,fex0)
forcoth[0:18] = forcoth[0:18] + (1/17)*(fex1-fex0)*(t[0:18]-1)
forcoth[18:NT] = forcoth[18:NT] + (fex1-fex0)

# Optimal long-run savings rate used for transversality (Question) (?)
optlrsav = (delk + .004)/(delk + .004*alpha + rho)*gamma

cost1 = np.zeros(NT)
cumetree = np.zeros(NT)
cumetree[0] = 100
cpricebase = cprice0*(1+gcprice)**(5*(t-1))

@njit('(float64[:], int32)')
def InitializeLabor(il,iNT):
    for i in range(1,iNT):
        il[i] = il[i-1]*(popasym / il[i-1])**popadj

@njit('(float64[:], int32)')
def InitializeTFP(ial,iNT):
    for i in range(1,iNT):
        ial[i] = ial[i-1]/(1-ga[i-1])

@njit('(float64[:], int32)')
def InitializeGrowthSigma(igsig,iNT):
    for i in range(1,iNT):
        igsig[i] = igsig[i-1]*((1+delsig)**tstep)

@njit('(float64[:], float64[:], float64[:], int32)')
def InitializeSigma(isigma,igsig,icost1,iNT):
```

```
    for i in range(1,iNT):
        isigma[i] =  isigma[i-1] * np.exp(igsig[i-1] * tstep)
        icost1[i] = pbacktime[i] * isigma[i]  / expcost2 /1000

@njit('(float64[:], int32)')
def InitializeCarbonTree(icumetree,iNT):
    for i in range(1,iNT):
        icumetree[i] = icumetree[i-1] + etree[i-1]*(5/3.666)


"""
Functions of the model
"""

"""
First: Functions related to emissions of carbon and weather damages
"""

# Determines the emission of carbon by industry;
@njit('float64(float64[:],float64[:],float64[:],int32)')
def fEIND(iYGROSS, iMIU, isigma,index):
    return isigma[index] * iYGROSS[index] * (1 - iMIU[index])

# Retuns the total carbon emissions;
@njit('float64(float64[:],int32)')
def fE(iEIND,index):
    return iEIND[index] + etree[index]

# Cumulative industrial emission of carbon;
@njit('float64(float64[:],float64[:],int32)')
def fCCA(iCCA,iEIND,index):
    return iCCA[index-1] + iEIND[index-1] * 5 / 3.666

# Cumulative total carbon emission;
@njit('float64(float64[:],float64[:],int32)')
def fCCATOT(iCCA,icumetree,index):
    return iCCA[index] + icumetree[index]

# Dynamics of the radiative forcing;
@njit('float64(float64[:],int32)')
def fFORC(iMAT,index):
    return fco22x * np.log(iMAT[index]/588.000)/np.log(2) + forcoth[index]

# Dynamics of Omega (Damage function);
@njit('float64(float64[:],int32)')
def fDAMFRAC(iTATM,index):
    return a1*iTATM[index] + a2*iTATM[index]**a3

# Calculate damages as a function of Gross industrial production;
@njit('float64(float64[:],float64[:],int32)')
def fDAMAGES(iYGROSS,iDAMFRAC,index):
    return iYGROSS[index] * iDAMFRAC[index]

# Dynamics of Lambda - cost of the reduction of carbon emission (Abatement
    cost);
@njit('float64(float64[:],float64[:],float64[:],int32)')
def fABATECOST(iYGROSS,iMIU,icost1,index):
    return iYGROSS[index] * icost1[index] * iMIU[index]**expcost2

# Marginal Abatement cost;
@njit('float64(float64[:],int32)')
def fMCABATE(iMIU,index):
    return pbacktime[index] * iMIU[index]**(expcost2-1)
```

```
# Price of carbon reduction;
@njit('float64(float64[:],int32)')
def fCPRICE(iMIU,index):
    return pbacktime[index] * (iMIU[index])**(expcost2-1)

# Dynamics of the carbon concentration in the atmosphere;
@njit('float64(float64[:],float64[:],float64[:],int32)')
def fMAT(iMAT,iMUP,iE,index):
    if(index == 0):
        return mat0
    else:
        return iMAT[index-1]*b11 + iMUP[index-1]*b21 + iE[index-1] * 5 /
            3.666

# Dynamics of the carbon concentration in the ocean LO level;
@njit('float64(float64[:],float64[:],int32)')
def fMLO(iMLO,iMUP,index):
    if(index == 0):
        return mlo0
    else:
        return iMLO[index-1] * b33  + iMUP[index-1] * b23

# Dynamics of the carbon concentration in the ocean UP level;
@njit('float64(float64[:],float64[:],float64[:],int32)')
def fMUP(iMAT,iMUP,iMLO,index):
    if(index == 0):
        return mup0
    else:
        return iMAT[index-1]*b12 + iMUP[index-1]*b22 + iMLO[index-1]*b32

# Dynamics of the atmospheric temperature;
@njit('float64(float64[:],float64[:],float64[:],int32)')
def fTATM(iTATM,iFORC,iTOCEAN,index):
    if(index == 0):
        return tatm0
    else:
        return iTATM[index-1] + c1 * (iFORC[index] - lam * iTATM[index-1]
            - c3 * (iTATM[index-1] - iTOCEAN[index-1]))

# Dynamics of the ocean temperature;
@njit('float64(float64[:],float64[:],int32)')
def fTOCEAN(iTATM,iTOCEAN,index):
    if(index == 0):
        return tocean0
    else:
        return iTOCEAN[index-1] + c4 * (iTATM[index-1] - iTOCEAN[index-1])

"""
Second: Function related to economic variables
"""

# Total production without climate losses (YGROSS);
@njit('float64(float64[:],float64[:],float64[:],int32)')
def fYGROSS(ial,il,iK,index):
    return ial[index] * ((il[index]/1000)**(1-gamma)) * iK[index]**gamma

# Production under the climate damages cost;
@njit('float64(float64[:],float64[:],int32)')
def fYNET(iYGROSS, iDAMFRAC, index):
    return iYGROSS[index] * (1 - iDAMFRAC[index])

# Production after abatement cost;
```

```
@njit('float64(float64[:],float64[:],int32)')
def fY(iYNET,iABATECOST,index):
    return iYNET[index] - iABATECOST[index]

# Consumption;
@njit('float64(float64[:],float64[:],int32)')
def fC(iY,iI,index):
    return iY[index] - iI[index]

# Per capita consumption;
@njit('float64(float64[:],float64[:],int32)')
def fCPC(iC,il,index):
    return 1000 * iC[index] / il[index]

# Saving policy: investment
@njit('float64(float64[:],float64[:],int32)')
def fI(iSAV,iY,index):
    return iSAV[index] * iY[index]

# Capital dynamics Eq. 13
@njit('float64(float64[:],float64[:],int32)')
def fK(iK,iI,index):
    if(index == 0):
        return k0
    else:
        return (1-delk)**tstep * iK[index-1] + tstep * iI[index-1]

# Interest rate equation;
@njit('float64(float64[:],int32)')
def fRI(iCPC,index):
    return (1 + rho) * (iCPC[index+1]/iCPC[index])**(alpha/tstep) - 1

# Periodic utility: A form of Eq. 2
@njit('float64(float64[:],float64[:],int32)')
def fCEMUTOTPER(iPERIODU,il,index):
    return iPERIODU[index] * il[index] * rr[index]

# The term between brackets in Eq. 2
@njit('float64(float64[:],float64[:],int32)')
def fPERIODU(iC,il,index):
    return ((iC[index]*1000/il[index])**(1-alpha) - 1) / (1 - alpha) - 1

# Utility function
@guvectorize([(float64[:], float64[:])], '(n),␣(m)')
def fUTILITY(iCEMUTOTPER, resUtility):
    resUtility[0] = tstep * scale1 * np.sum(iCEMUTOTPER) + scale2


#Ising Opinion Mechanism Function

def Ising(T,Iex,mc_steps):

    #breaking the Aij and Sij matrices into local groups
    uS = unpack_matrix(Sij,Lgx,Lgx)
    uA = unpack_matrix(Aij,Lgx,Lgx)

    for tt in range(mc_steps):

        #local field contribution
        for i in range(L):
            for j in range(L):

                hij = 0
```

```
                iisum = 0
                losum = 0

                neigh1 = list(G.neighbors((i,j)))
                for z in range(len(neigh1)):

                    #index selection
                    tup1 = neigh1[z]
                    z1 = tup1[0]
                    z2 = tup1[1]

                    #contribution of interpersonal connections
                    iisum += Aij[z1,z2]*Sij[z1,z2]

                #local group identification
                ix = np.floor(i/Lg)
                jx = np.floor(j/Lg)

                #contribution of local group
                losum = np.sum(uS[(ix, jx)]*uA[(ix, jx)])

                hij = (1/kij[i,j])*(iisum+(losum/Ng))+Iex

                #opinion switch mechanism
                if T == 0:

                    if hij*Sij[i,j] >= 0:

                        pij = 0

                    else:

                        pij = (1-Aij[i,j])
                        Proll = np.random.random()

                        if Proll <= pij:

                            Sij[i,j] = (-1)*Sij[i,j]

                else:

                    if hij*Sij[i,j] > 0:

                        pij = (1-Aij[i,j])*np.exp((-hij*Sij[i,j])/T)
                        Proll = np.random.random()

                        if Proll <= pij:

                            Sij[i,j] = (-1)*Sij[i,j]

                    else:

                        pij = (1-Aij[i,j])*(1-np.exp((hij*Sij[i,j])/T))
                        Proll = np.random.random()

                        if Proll <= pij:

                            Sij[i,j] = (-1)*Sij[i,j]

    return np.sum(Sij)/N
```

```
# Variable list initialization

OPINION = np.zeros(NT)  # opinion results of network
COUPLING = np.zeros(NT)  # coupling equation

K = np.zeros(NT)
YGROSS = np.zeros(NT)
EIND = np.zeros(NT)
E = np.zeros(NT)
CCA = np.zeros(NT)
CCATOT = np.zeros(NT)
MAT = np.zeros(NT)
MLO = np.zeros(NT)
MUP = np.zeros(NT)
FORC = np.zeros(NT)
TATM = np.zeros(NT)
TOCEAN = np.zeros(NT)
DAMFRAC = np.zeros(NT)
DAMAGES = np.zeros(NT)
ABATECOST = np.zeros(NT)
MCABATE = np.zeros(NT)
CPRICE = np.zeros(NT)
YNET = np.zeros(NT)
Y = np.zeros(NT)
I = np.zeros(NT)
C = np.zeros(NT)
CPC = np.zeros(NT)
RI = np.zeros(NT)
PERIODU = np.zeros(NT)
CEMUTOTPER = np.zeros(NT)

InitializeLabor(l,NT)
InitializeTFP(al,NT)
InitializeGrowthSigma(gsig,NT)
InitializeSigma(sigma,gsig,cost1,NT)
InitializeCarbonTree(cumetree,NT)

#Network parameters

T = 0.15          #temperature
Iex = 0.0         #external stimulation
mc_steps = 100    #MC steps

#Model equations with fixed control variables

MIU = np.zeros(NT)       #abatement array
peak_T = np.zeros(NT)       #peak temperature array
SAV = 0.23*np.ones(NT)      #saving rate fixed at 0.23

MIU[0] = miu0
peak_T[0] = 2.5  # initialization peak temperature - Paris Agreement
    target

#Party-specific optimization target variables
MIU_g = np.zeros(NT)       # abatement array for Green party
MIU_r = np.zeros(NT)       # abatement array for Lukewarmer party

peak_T_g = np.zeros(NT)       # Green party peak temperature projection
peak_T_r = np.zeros(NT)       # Lukewarmer party peak temperature
    projection

MIU_g[0] = 0
MIU_r[0] = 0
```

```
peak_T_g[0] = 0
peak_T_r[0] = 0

for i in range(NT):

        K[i] = fK(K,I,i)
        YGROSS[i] = fYGROSS(al,l,K,i)
        EIND[i] = fEIND(YGROSS, MIU, sigma,i)
        E[i] = fE(EIND,i)
        CCA[i] = fCCA(CCA,EIND,i)
        CCATOT[i] = fCCATOT(CCA,cumetree,i)
        MAT[i] = fMAT(MAT,MUP,E,i)
        MLO[i] = fMLO(MLO,MUP,i)
        MUP[i] = fMUP(MAT,MUP,MLO,i)
        FORC[i] = fFORC(MAT,i)
        TATM[i] = fTATM(TATM,FORC,TOCEAN,i)
        TOCEAN[i] = fTOCEAN(TATM,TOCEAN,i)
        DAMFRAC[i] = fDAMFRAC(TATM,i)
        DAMAGES[i] = fDAMAGES(YGROSS,DAMFRAC,i)
        ABATECOST[i] = fABATECOST(YGROSS,MIU,cost1,i)
        MCABATE[i] = fMCABATE(MIU,i)
        CPRICE[i] = fCPRICE(MIU,i)
        YNET[i] = fYNET(YGROSS,DAMFRAC,i)
        Y[i] = fY(YNET,ABATECOST,i)
        I[i] = fI(SAV,Y,i)
        C[i] = fC(Y,I,i)
        CPC[i] = fCPC(C,l,i)
        PERIODU[i] = fPERIODU(C,l,i)
        CEMUTOTPER[i] = fCEMUTOTPER(PERIODU,l,i)
        RI[i] = fRI(CPC,i)

        #Coupling Equation

        tax_t = (MCABATE[i]*E[i]/l[i])*(1/1000)    # USD per person
        dam_t = DAMAGES[i]/l[i]*(1/1000000)        # USD per person

        COUPLING[i] = (1)*dam_t - tax_t + (0.001)*np.sqrt(peak_T[i])

        Iex = 100*COUPLING[i] # free, uncalibrated external influence
            parameter coupling

        #Network Opinion Check
        OPINION[i] = Ising(T,Iex,mc_steps)

        if i == NT-1:
            break

        else:

            #Optimization Schemes (for both parties)

            #GREENS
            MIU_g[i+1],peak_T_g[i+1] = DICE_optimizer(i,5.0,l[i],al[i],
                gsig[i],sigma[i],
                cumetree[i],MIU[i],K[i],YGROSS[i],EIND[i],E[i],CCA[i],
                    CCATOT[i],MAT[i],
                MLO[i],MUP[i],FORC[i],TATM[i],TOCEAN[i],DAMFRAC[i],DAMAGES
                    [i],ABATECOST[i],
                MCABATE[i],CPRICE[i],YNET[i],Y[i],I[i],C[i],CPC[i],RI[i],
                    PERIODU[i],CEMUTOTPER[i])

            #LUKEWARMERS
```

```
            MIU_r[i+1],peak_T_r[i+1] = DICE_optimizer(i,2.0,l[i],al[i],
                gsig[i],sigma[i],
                cumetree[i],MIU[i],K[i],YGROSS[i],EIND[i],E[i],CCA[i],
                    CCATOT[i],MAT[i],
                MLO[i],MUP[i],FORC[i],TATM[i],TOCEAN[i],DAMFRAC[i],DAMAGES
                    [i],ABATECOST[i],
                MCABATE[i],CPRICE[i],YNET[i],Y[i],I[i],C[i],CPC[i],RI[i],
                    PERIODU[i],CEMUTOTPER[i])

            #Peak temperature projection

            peak_T[i+1] = 0.5*peak_T_g[i+1]+0.5*peak_T_r[i+1]


            #Political party selection (parameters)
            if OPINION[i] > 0:

                MIU[i+1] = MIU_g[i+1]

            else:

                MIU[i+1] = MIU_r[i+1]


TT = np.linspace(2015, 2115, 21, dtype = np.int32)

def PlotFigures():

    pos_OPINION = OPINION.copy()
    neg_OPINION = OPINION.copy()

    pos_OPINION[pos_OPINION <= 0] = np.nan
    neg_OPINION[neg_OPINION > 0] = np.nan

    figOPINION = plt.figure(figsize=(8,6))
    plt.plot(TT,OPINION,'k--',alpha=0.5)
    plt.plot(TT,pos_OPINION,'g.')
    plt.plot(TT,neg_OPINION,'r.')
    figOPINION.suptitle('Opinion␣average', fontsize=14)
    plt.grid(alpha=0.5)
    plt.xlabel('Years', fontsize=12)
    plt.ylabel('$<S>$', fontsize=12)

    figTATM = plt.figure(figsize=(8,6))
    plt.plot(TT,TATM,'b.')
    plt.plot(TT,TATM,'b-',alpha=0.5)
    figTATM.suptitle('Increase␣temperature␣of␣the␣atmosphere␣(TATM)',
        fontsize=14)
    plt.grid(alpha=0.5)
    plt.xlabel('Years', fontsize=12)
    plt.ylabel('Degrees␣C␣from␣1900', fontsize=12)

    figTOCEAN = plt.figure(figsize=(8,6))
    plt.plot(TT,TOCEAN,'b.')
    plt.plot(TT,TOCEAN,'b-',alpha=0.5)
    figTOCEAN.suptitle('Increase␣temperature␣of␣the␣ocean␣(TOCEAN)',
        fontsize=14)
    plt.grid(alpha=0.5)
    plt.xlabel('Years', fontsize=12)
    plt.ylabel('Degrees␣C␣from␣1900', fontsize=12)

    figMU = plt.figure(figsize=(8,6))
    plt.plot(TT,MUP,'b.')
```

```
plt.plot(TT,MUP,'b-',alpha=0.5)
figMU.suptitle('Carbon␣concentration␣increase␣in␣shallow␣oceans␣(MU)',
     fontsize=14)
plt.grid(alpha=0.5)
plt.xlabel('Years', fontsize=12)
plt.ylabel('GtC␣from␣1750', fontsize=12)

figML = plt.figure(figsize=(8,6))
plt.plot(TT,MLO,'b.')
plt.plot(TT,MLO,'b-',alpha=0.5)
figML.suptitle('Carbon␣concentration␣increase␣in␣lower␣oceans␣(ML)',
    fontsize=14)
plt.grid(alpha=0.5)
plt.xlabel('Years', fontsize=12)
plt.ylabel('GtC␣from␣1750', fontsize=12)

figDAM = plt.figure(figsize=(8,6))
plt.plot(TT,DAMAGES,'b.')
plt.plot(TT,DAMAGES,'b-',alpha=0.5)
figDAM.suptitle('Damages', fontsize=14)
plt.grid(alpha=0.5)
plt.xlabel('Years', fontsize=12)
plt.ylabel('trillions␣2010␣USD␣per␣year', fontsize=12)

figDAMFRAC = plt.figure(figsize=(8,6))
plt.plot(TT,DAMFRAC,'b.')
plt.plot(TT,DAMFRAC,'b-',alpha=0.5)
figDAMFRAC.suptitle('Damages␣as␣fraction␣of␣gross␣output', fontsize
    =14)
plt.grid(alpha=0.5)
plt.xlabel('Years', fontsize=12)
plt.ylabel('', fontsize=12)

figCOSTRED = plt.figure(figsize=(8,6))
plt.plot(TT,ABATECOST,'b.')
plt.plot(TT,ABATECOST,'b-',alpha=0.5)
figCOSTRED.suptitle('Cost␣of␣emissions␣reductions', fontsize=14)
plt.grid(alpha=0.5)
plt.xlabel('Years', fontsize=12)
plt.ylabel('trillions␣2010␣USD␣per␣year', fontsize=12)

figMarg = plt.figure(figsize=(8,6))
plt.plot(TT,MCABATE,'b.')
plt.plot(TT,MCABATE,'b-',alpha=0.5)
figMarg.suptitle('Marginal␣abatement␣cost', fontsize=14)
plt.grid(alpha=0.5)
plt.xlabel('Years', fontsize=12)
plt.ylabel('2010␣USD␣per␣ton␣CO2', fontsize=12)

figMIU = plt.figure(figsize=(8,6))
plt.plot(TT,MIU,'b.')
plt.plot(TT,MIU,'b-',alpha=0.5)
figMIU.suptitle('Carbon␣emission␣control␣rate', fontsize=14)
plt.grid(alpha=0.5)
plt.xlabel('Years', fontsize=12)
plt.ylabel('Rate', fontsize=12)

figE = plt.figure(figsize=(8,6))
plt.plot(TT,E,'b.')
plt.plot(TT,E,'b-',alpha=0.5)
figE.suptitle('Total␣CO2␣emission', fontsize=14)
plt.grid(alpha=0.5)
plt.xlabel('Years', fontsize=12)
```

```
plt.ylabel('GtCO2 per year', fontsize=12)

figMAT = plt.figure(figsize=(8,6))
plt.plot(TT,MAT,'b.')
plt.plot(TT,MAT,'b-',alpha=0.5)
figMAT.suptitle('Carbon concentration increase in the atmosphere',
    fontsize=14)
plt.grid(alpha=0.5)
plt.xlabel('Years', fontsize=12)
plt.ylabel('GtC from 1750', fontsize=12)

figFORC = plt.figure(figsize=(8,6))
plt.plot(TT,FORC,'b.')
plt.plot(TT,FORC,'b-',alpha=0.5)
figFORC.suptitle('Increase in radiative forcing', fontsize=14)
plt.grid(alpha=0.5)
plt.xlabel('Years', fontsize=12)
plt.ylabel('watts per m2 from 1900', fontsize=12)

figC = plt.figure(figsize=(8,6))
plt.plot(TT,C,'b.')
plt.plot(TT,C,'b-',alpha=0.5)
figC.suptitle('Consumption', fontsize=14)
plt.grid(alpha=0.5)
plt.xlabel('Years', fontsize=12)
plt.ylabel('trillions 2010 USD per year', fontsize=12)

figY = plt.figure(figsize=(8,6))
plt.plot(TT,Y,'b.')
plt.plot(TT,Y,'b-',alpha=0.5)
figY.suptitle('Gross product net of abatement and damages', fontsize
    =14)
plt.grid(alpha=0.5)
plt.xlabel('Years', fontsize=12)
plt.ylabel('trillions 2010 USD per year', fontsize=12)

figYGROSS = plt.figure(figsize=(8,6))
plt.plot(TT,YGROSS,'b.')
plt.plot(TT,YGROSS,'b-',alpha=0.5)
figYGROSS.suptitle('World gross product', fontsize=14)
plt.grid(alpha=0.5)
plt.xlabel('Years', fontsize=12)
plt.ylabel('trillions 2010 USD per year', fontsize=12)

figYGROSSbyY = plt.figure(figsize=(8,6))
plt.plot(TT,YGROSS-Y,'b.')
plt.plot(TT,YGROSS-Y,'b-',alpha=0.5)
figYGROSSbyY.suptitle('Abatement and damages costs', fontsize=14)
plt.grid(alpha=0.5)
plt.xlabel('Years', fontsize=12)
plt.ylabel('trillions 2010 USD per year', fontsize=12)

figSAV = plt.figure(figsize=(8,6))
plt.plot(TT,SAV,'b.')
plt.plot(TT,SAV,'b-',alpha=0.5)
figSAV.suptitle('Saving rate', fontsize=14)
plt.grid(alpha=0.5)
plt.xlabel('Years', fontsize=12)
plt.ylabel('rate', fontsize=12)

figI = plt.figure(figsize=(8,6))
plt.plot(TT,I,'b.')
plt.plot(TT,I,'b-',alpha=0.5)
```

```
    figI.suptitle('Investment (I)', fontsize=14)
    plt.grid(alpha=0.5)
    plt.xlabel('Years', fontsize=12)
    plt.ylabel('trillions 2010 USD per year', fontsize=12)

    figpT = plt.figure(figsize=(8,6))
    plt.plot(TT,peak_T,'b.')
    plt.plot(TT,peak_T,'b-',alpha=0.5)
    figpT.suptitle('Peak Temperature Projection', fontsize=14)
    plt.grid(alpha=0.5)
    plt.xlabel('Years', fontsize=12)
    plt.ylabel('Degrees C from 1900', fontsize=12)

    plt.show()


def DataSave():

    np.save("OPINION.npy", OPINION)
    np.save("l.npy", l)
    np.save("al.npy", al)
    np.save("gsig.npy", gsig)
    np.save("TATM.npy", TATM)
    np.save("TOCEAN.npy", TOCEAN)
    np.save("MUP.npy", MUP)
    np.save("MLO.npy", MLO)
    np.save("DAMAGES.npy", DAMAGES)
    np.save("DAMFRAC.npy", DAMFRAC)
    np.save("ABATECOST.npy", ABATECOST)
    np.save("MCABATE.npy", MCABATE)
    np.save("MIU.npy", MIU)
    np.save("E.npy", E)
    np.save("MAT.npy", MAT)
    np.save("FORC.npy", FORC)
    np.save("C.npy", C)
    np.save("Y.npy", Y)
    np.save("YGROSS.npy", YGROSS)
    np.save("SAV.npy", SAV)
    np.save("I.npy", I)
    np.save("COUPLING.npy", COUPLING)
    np.save("MIU_g.npy", MIU_g)
    np.save("MIU_r.npy", MIU_r)
    np.save("peak_T_g.npy", peak_T_g)
    np.save("peak_T_r.npy", peak_T_r)


figLAB = plt.figure(figsize=(8,6))
plt.plot(TT,l,'b.')
plt.plot(TT,l,'b-',alpha=0.5)
figLAB.suptitle('Population growth over time ($L$)', fontsize=14)
plt.grid(alpha=0.5)
plt.xlabel('Years', fontsize=12)
plt.ylabel('Population (million)', fontsize=12)

figTFP = plt.figure(figsize=(8,6))
plt.plot(TT,al,'b.')
plt.plot(TT,al,'b-',alpha=0.5)
figTFP.suptitle('Total factor productivity ($A$)', fontsize=14)
plt.grid(alpha=0.5)
plt.xlabel('Years', fontsize=12)
plt.ylabel('TFP', fontsize=12)

figGSIG = plt.figure(figsize=(8,6))
```

```
plt.plot(TT,gsig,'b.')
plt.plot(TT,gsig,'b-',alpha=0.5)
figGSIG.suptitle('Growth of sigma', fontsize=14)
plt.grid(alpha=0.5)
plt.xlabel('Years', fontsize=12)
plt.ylabel('$g_sigma$', fontsize=12)


PlotFigures()
#DataSave()

print(f'Computation time = {datetime.now() - startTime}')
```

Bibliography

[1] James E. Lovelock. "Gaia as seen through the atmosphere". In: *Atmospheric Environment* 6.8 (1967), pp. 579–580.

[2] James E. Lovelock and Lynn Margulis. "Atmospheric homeostasis by and for the biosphere: the gaia hypothesis". In: *Tellus* 26.1-2 (1974), pp. 2–10.

[3] John Cook et al. "Quantifying the consensus on anthropogenic global warming in the scientific literature". In: *Environmental Research Letters* 8.2 (2013), p. 024024.

[4] John Cook et al. "Consensus on consensus: a synthesis of consensus estimates on human-caused global warming". In: *Environmental Research Letters* 11.4 (2016), p. 048002.

[5] Ernst Ising. "Contribution to the Theory of Ferromagnetism". In: *Z. Phys.* 31 (1925), pp. 253–258.

[6] M. E. J. Newman and G. T. Barkema. *Monte Carlo methods in statistical physics*. Clarendon Press, 1999.

[7] P. Erdos and A. Rényi. "On Random Graphs". In: *Publicationes Mathematicae* 6 (1959), pp. 290–297.

[8] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. "Exploring Network Structure, Dynamics, and Function using NetworkX". In: *Proceedings of the 7th Python in Science Conference*. Pasadena, CA USA, 2008, pp. 11–15.

[9] David Lazer et al. "Computational Social Science". In: *Science* 323.5915 (2009), pp. 721–723.

[10] Stanley Milgram. "The Small World Problem". In: *Psychology Today* 2 (Jan. 1967).

[11] Manfred Kochen. *The Small World*. 1989.

[12] Duncan Watts and Steven Strogatz. "Collective Dynamics of Small World Networks". In: *Nature* 393 (July 1998), pp. 440–2.

[13] Duncan J. Watts and Steven H. Strogatz. "Collective dynamics of 'small-world' networks". In: *Nature* 393.6684 (1998), pp. 440–442.

[14] Reka Albert, Hawoong Jeong, and Albert-László Barabási. "Diameter of the World-Wide Web". In: *Nature* 401 (Sept. 1999), pp. 130–131.

[15] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. "On Power-Law Relationships of the Internet Topology". In: *ACM SIGCOMM Computer Communication Review* 29 (1999).

[16] Hawoong Jeong et al. "The Large-Scale Organization of Metabolic Networks". In: *Nature* 407 (Nov. 2000), pp. 651–4.

[17] Albert-László Barabási and Reka Albert. "Emergence of Scaling in Random Networks." In: *Science (New York, N.Y.)* 286 (Nov. 1999), pp. 509–12.

[18] B. Alder and T. Wainwright. "Phase Transition for a Hard Sphere System". In: *jcph* 27 (Nov. 1957), pp. 1208–.

[19] B.J. Alder and T.E. Wainwright. "Studies in Molecular Dynamics. I. General Method". In: *jcph* 31 (Aug. 1959), pp. 459–.

[20] Metropolis et al. "Equation of state calculations for fast computing machines". In: *Journal of Chemical Physics 6* 21 (Jan. 1953), pp. 1087–.

[21] W. Weidlich. "The statistical description of polarization phenomena in society". In: *British Journal of Mathematical and Statistical Psychology* 24.2 (1971), pp. 251–266.

[22] Serge Galam, Yuval Gefen (Feigenblat), and Yonathan Shapir. "Sociophysics: A new approach of sociological collective behaviour. I. mean-behaviour description of a strike". In: *The Journal of Mathematical Sociology* 9.1 (1982), pp. 1–13.

[23] Serge Galam and Serge Moscovici. "Towards a Theory of Collective Phenomena: Consensus and Attitude Changes in Groups". In: *European Journal of Social Psychology* 21 (Jan. 1991), pp. 49–74.

[24] Anthony Kleerekoper, Adam Coates, and Liangxiu Han. "A Unified Framework for Opinion Dynamics". In: July 2018.

[25] Dietrich Stauffer. "Opinion Dynamics and Sociophysics". In: *Encyclopedia of Complexity and Systems Science* (June 2007).

[26] Dietrich Stauffer et al. *Biology, Sociology, Geology by Computational Physicists (Monograph Series on Nonlinear Science and Complexity)*. Jan. 2006.

[27] Peter Clifford and Aidan Sudbury. "A Model for Spatial Conflict". In: *Biometrika* 60 (Dec. 1973).

[28] Richard Holley and Thomas Liggett. "Ergodic Theorems for Weakly Interacting Infinite Systems and the Voter Model". In: *Ann. Probability* 3 (Aug. 1975).

[29] Claudio Castellano, Miguel Muñoz, and Romualdo Pastor-Satorras. "The nonlinear q-voter model". In: *Physical review. E, Statistical, nonlinear, and soft matter physics* 80 (Oct. 2009).

[30] Serge Galam. "Minority Opinion Spreading in Random Geometry". In: *Physics of Condensed Matter* 25 (Mar. 2002).

[31] Katarzyna Sznajd-Weron and Jozef Sznajd. "Opinion Evolution in Closed Community". In: *International Journal of Modern Physics C* 11 (Jan. 2001).

[32] Giulio Rossetti, Letizia Milli, and Salvatore Rinzivillo. "NDlib: A Python Library to Model and Analyze Diffusion Processes over Complex Networks". In: Apr. 2018, pp. 183–186.

[33] Andrzej Grabowski and R.A. Kosiński. "Ising-based model of opinion formation in a complex network of interpersonal interactions". In: *Physica A: Statistical Mechanics and its Applications* 361 (Mar. 2006), 651–664.

[34] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. "Gephi: An Open Source Software for Exploring and Manipulating Networks". In: (2009). URL: https://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154.

[35] William Nordhaus. "The 'DICE' Model: Background and Structure of a Dynamic Integrated Climate-Economy Model of the Economics of Global Warming". In: (Mar. 1992).

[36] William Nordhaus and Joseph Boyer. *Warming the World: Economic Models of Global Warming*. Jan. 2000.

[37] William Nordhaus. "Revisiting the social cost of carbon". In: *Proceedings of the National Academy of Sciences* 114 (Jan. 2017).

[38] Robert Solow. *Growth Theory: An Exposition*. Vol. 22. Jan. 1971.

[39] Claudia Wieners. *God does not play DICE – but Bill Nordhaus does! What can models tell us about the economics of climate change?* Dec. 2018. URL: https://blogs.egu.eu/divisions/cl/2018/12/03/god-does-not-play-dice-but-bill-nordhaus-does-what-can-models-tell-us-about-the-economics-of-climate-change/.

[40] L.G. Epstein and S.E. Zin. "Substitution, risk aversion, and the intertemporal behavior of consumption and asset returns: a theoretical framework". In: *Econometrica* 57 (Jan. 1989), pp. 937–969.

[41] Larry Epstein and Stanley Zin. "Substitution, Risk Aversion, and the Temporal Behavior of Consumption and Asset Returns: An Empirical Analysis". In: *Journal of Political Economy* 99 (Feb. 1991), pp. 263–86.

[42] United Nations, Department Affairs, and Population Division. *World Population Prospects 2019*. 2019.

[43] Richard Tol. "The Economic Effects of Climate Change". In: *Journal of Economic Perspectives* 23 (Apr. 2009), pp. 29–51.

[44] Richard Tol. "The Economic Effects of Climate Change". In: *The Journal of Economic Perspectives* 28 (May 2014).

[45] IPCC. "Climate Change 2014: Synthesis Report. Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change". In: *IPCC, Geneva, Switzerland* (2014), p. 151. URL: https://www.ipcc.ch/report/ar5/syr/.

[46] Reto Knutti and Gabriele Hegerl. "The Equilibrium Sensitivity of the Earth's Temperature to Radiation Changes." In: *Nature Geoscience* 1 (Oct. 2008), pp. 735–743.

[47] Nicholas Stern. *The Economics of Climate Change: The Stern Review*. Cambridge University Press, 2007.

[48] Cara A. Horowitz. "Paris Agreement". In: *International Legal Materials* 55.4 (2016), 740–755.

[49] Sylvie Geisendorf. "Evolutionary Climate-Change Modelling: A Multi-Agent Climate-Economic Model". In: *Computational Economics* 52 (Sept. 2017).

[50] Martin Weitzman. "What is the "Damages Function" for Global Warming – and What Difference Might It Make?" In: *Climate Change Economics (CCE)* 01 (May 2010), pp. 57–69.

[51] Jeroen van den Bergh. "How sensitive is Nordhaus to Weitzman? Climate policy in DICE with an alternative damage function". In: *Economics Letters* 117 (Oct. 2012), pp. 372–374.

[52] Antonin Pottier et al. "The Comparative Impact of Integrated Assessment Models' Structures on Optimal Mitigation Policies". In: *Environmental Modeling and Assessment* (Feb. 2015).

[53] Tarek Hassan et al. "The Global Impact of Brexit Uncertainty". In: *Institute for New Economic Thinking Working Paper Series* (Dec. 2019), pp. 1–60.