

ROBUSTNESS OF TURING PATTERNS ON NETWORKS

CASPER H. L. BEENTJES

Mathematical Institute, University of Oxford, Oxford, UK

ABSTRACT. Self-organising patterns originating from the Turing instability have been widely studied in the context of continuous media. One commonly studied question is how robust these Turing patterns are to a variety of external and internal changes to the underlying system and dynamics. More recently the extension of Turing's work to the case of networks has been made, introducing the element of the network topology to Turing pattern formation. This introduces a new kind of robustness question, how robust are the developing patterns to changes in the network topology. We address this question by looking at pattern change under a bond percolation process. In this paper we also explain some of the theory of Turing patterns on networks, the parameter constraints needed for such patterns and its influence on the final long-time patterns that are observed.

1. INTRODUCTION

The spontaneous emergence of patterns out of a disordered initial state is widely observed within models in physics, chemistry, biology and ecology [7, 15, 21, 25]. These self-organising patterns have attracted a lot of attention from the scientific community and have for example been coined as a mechanism for cell differentiation in the developmental stage of organisms [24]. Seminal work from 1952 by Alan Turing showed that it is possible to create spontaneous pattern formation in the mathematical framework of reaction-diffusion equations [23], now known as Turing patterns. His work revolves around a partial differential equation formalism and it is an intricate diffusion difference between at least two species that can generate spatially extended patterns, whereas in the absence of this diffusion a homogeneous state would be observed.

The original Turing pattern formalism is necessarily a continuous medium description and as such is therefore insufficient to describe self-organisation in cases where there is no inherent continuous space. Examples can be found in the early stages of biological development, where individual cells make up discrete building blocks of space over which chemicals spread out [19], or in ecology, where the metapopulation description is used to describe fragmented habitats [9]. In many such discrete models the dynamics of chemicals, animals, or other agents, can be described by species residing on network nodes which can be transported between nodes over the network edges. In 1971 Othmer and Scriven showed how to extend the Turing pattern formalism to dynamics on networks [19]. In this network formalism the topology of the network provides a new ingredient in the study of pattern formation. Initially, however, this newly developed theory was only applied to a small portion of possible networks, namely lattices and small networks [10, 14, 19, 20].

In the last decade Turing patterns on more general networks became topic of scientific study, initiated by a paper by Nakao and Mikhailov [17], who for the first time considered Turing patterns

E-mail address: beentjes@maths.ox.ac.uk.

Date: Hillary term 2017.

2010 Mathematics Subject Classification. 34D20, 92C15.

This text is based on a technical report for the Hilary 2017 Oxford Mathematics course C5.4 Networks lectured by Prof. H. Harrington.

on large complex networks. This opened the door to the usage of large unstructured (random) networks, scale-free networks and small world models, all widely studied in networks field. This extends the usage of Turing patterns to cases where the networks in the model are not necessarily derived from some consideration of physical space, e.g. functional areas in the brain and their connections [22]. The theory has since then been extended to directed networks [4], stochastic Turing patterns [5] and multilayer networks [2, 11], in some cases showing distinctively different behaviour compared to the classical Turing patterns.

Recently Asllani, Carletti and Fanelli studied the effects of changing the network topology on the creation and destruction of patterns [3]. They specifically considered two interacting species which were allowed to live on different networks, yielding an extra way to induce symmetry-breaking Turing patterns. A question inspired by the aforementioned work is how robust established Turing patterns on networks are, where we will have to define a notion of robustness. In a developmental biology scenario, for example, one might hope to have a Turing pattern developing even in the case where a (small) part of the original network is malfunctioning or missing, a problem known as percolation in the mathematics community. We therefore define robustness merely by considering whether a pattern forms or not. A more stringent robustness problem could be the question whether a specific pattern can persist in the case of a partially malfunctioning network. This, arguably harder, problem has been previously studied in the classical continuum framework, see for example Maini et al. [12]. In this paper, however, we focus on the former definition of robustness.

We first introduce the Turing instability in reaction-diffusion systems on networks in Section 2. One important difference between continuum and networked systems as noted by Nakao and Mikhailov is the form of the long-time patterns that form. In Section 3 we look at the (linearised) pattern space, which gives an idea of which patterns will form at certain points in our parameter space. This will turn out to be related to the robustness problem, which we will further discuss in Section 4. We will consider several (small) network models with equal number of nodes and see how robust patterns are on the respective networks.

2. TURING INSTABILITY ON NETWORKS

Reaction-diffusion models. For the sake of simplicity we consider the case with two interacting species U and V , whose concentration we denote by u and v respectively. More specifically we consider u to be an activator species, i.e. the production of u is autocatalytically stimulated by u , and v is an inhibitor, i.e. this species represses the production of u . In the absence of space, either in a network or continuous medium case, the dynamics of the species is then governed by the ordinary differential equations (ODEs)

$$(1a) \quad \frac{du}{dt} = f(u, v)$$

$$(1b) \quad \frac{dv}{dt} = g(u, v),$$

where the (local) interaction between u and v is captured by the functions $f(u, v)$ and $g(u, v)$. Turing in his 1952 paper added a continuous spatial component to the model and let the species diffuse, arriving at a reaction-diffusion system of the form

$$(2a) \quad \frac{\partial u}{\partial t} = f(u, v) + D_u \nabla^2 u$$

$$(2b) \quad \frac{\partial v}{\partial t} = g(u, v) + D_v \nabla^2 v,$$

where now D_u and D_v are the respective diffusion coefficients for u and v .

In this paper, however, we consider an undirected network G with N nodes ν_i for $i = 1, \dots, N$ and the usual adjacency matrix \mathbf{A} , i.e. $\mathbf{A}_{ij} = 1$ if there is a connection between ν_i and ν_j . The two species in our model are assumed to reside on the nodes ν_i of the graph, which we can represent

by the two vectors \mathbf{u} and \mathbf{v} , such that \mathbf{u}_i represents the concentration of species U on node ν_i . In the absence of a coupling between the different nodes this leads to dynamics similar to (1)

$$(3a) \quad \frac{d\mathbf{u}_i}{dt} = f(\mathbf{u}_i, \mathbf{v}_i)$$

$$(3b) \quad \frac{d\mathbf{v}_i}{dt} = g(\mathbf{u}_i, \mathbf{v}_i),$$

which now holds for $i = 1, \dots, N$. To add interaction between the species on different nodes Othmer and Scriven introduced Fickian diffusion between the nodes which leads to a system of N coupled reaction-diffusion ODEs

$$(4a) \quad \frac{d\mathbf{u}_i}{dt} = f(\mathbf{u}_i, \mathbf{v}_i) + D_u \sum_{j=1}^N \mathbf{A}_{ij}(\mathbf{u}_j - \mathbf{u}_i)$$

$$(4b) \quad \frac{d\mathbf{v}_i}{dt} = g(\mathbf{u}_i, \mathbf{v}_i) + D_v \sum_{j=1}^N \mathbf{A}_{ij}(\mathbf{v}_j - \mathbf{v}_i),$$

where we assume that the diffusion constants D_u and D_v can differ between the two species, but not across the network¹. If we define the matrix \mathbf{D} as the diagonal matrix with $\mathbf{D}_{ii} = k_i$, the vertex degree of ν_i , we can rewrite the diffusion part by using the graph Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$. We then define the vector functions $\mathbf{F}(\mathbf{u}, \mathbf{v})$ and $\mathbf{G}(\mathbf{u}, \mathbf{v})$ such that $\mathbf{F}_i(\mathbf{u}, \mathbf{v}) = f(\mathbf{u}_i, \mathbf{v}_i)$ and similarly for \mathbf{G} . With this notation we can rewrite the system of ODEs (4) in a form reminiscent of the original reaction-diffusion system (2) that Turing proposed

$$(5a) \quad \frac{d\mathbf{u}}{dt} = \mathbf{F}(\mathbf{u}, \mathbf{v}) - \varepsilon \mathbf{L}\mathbf{u}$$

$$(5b) \quad \frac{d\mathbf{v}}{dt} = \mathbf{G}(\mathbf{u}, \mathbf{v}) - \varepsilon \sigma \mathbf{L}\mathbf{v},$$

where we introduced $\varepsilon = D_u$ and $\sigma = D_v/D_u$ to bring notation in line with [17] and subsequent work. We note that the Laplacian matrix \mathbf{L} does differ in sign from the continuous Laplacian operator in the original system of equations (2), which is conventional in the networks literature.

Turing instability. The Turing instability classically manifests itself when a homogeneous steady state is stable in the absence of diffusion (i.e. when $D_u = D_v = 0$), but turns unstable when diffusion is added. This formalism carries over to the networks case. We assume that the reaction dynamics f and g allow a non-zero steady state (u^*, v^*) for (1). This yields a homogeneous steady state for the reaction-diffusion system (5) of the form $(\mathbf{u}^*, \mathbf{v}^*)$, which attains $\mathbf{u}_i = u^*$ and $\mathbf{v}_i = v^*$ at every node ν_i .

In the absence of diffusion, i.e. $\varepsilon = 0$, we demand $(\mathbf{u}^*, \mathbf{v}^*)$ to be a stable steady state, yielding the traditional conditions on the reaction dynamics

- (i) $f_u + g_v < 0$,
- (ii) $f_u g_v - f_v g_u > 0$,

where the above expressions are evaluated at (u^*, v^*) . If we now let $\varepsilon > 0$ the homogeneous steady state should become unstable. To derive the necessary conditions for this to happen we consider a (non-homogeneous) perturbation of the steady state, i.e. $\mathbf{u} = \mathbf{u}^* + \boldsymbol{\eta}_u$ and $\mathbf{v} = \mathbf{v}^* + \boldsymbol{\eta}_v$, such that

¹Such an anisotropy effect, however, could conceivably be accounted for by considering a weighted network and similarly adapted adjacency matrix \mathbf{A} . This has not (yet) been reported in the literature though.

$\|\boldsymbol{\eta}_u\|, \|\boldsymbol{\eta}_v\| \ll 1$. These perturbations then satisfy the following linearised ODEs

$$(6a) \quad \frac{d\boldsymbol{\eta}_u}{dt} = f_u \boldsymbol{\eta}_u + f_v \boldsymbol{\eta}_v - \varepsilon \mathbf{L} \boldsymbol{\eta}_u$$

$$(6b) \quad \frac{d\boldsymbol{\eta}_v}{dt} = g_u \boldsymbol{\eta}_u + g_v \boldsymbol{\eta}_v - \varepsilon \sigma \mathbf{L} \boldsymbol{\eta}_v,$$

with again all partial derivatives of f and g evaluated at the steady state. The perturbations $\boldsymbol{\eta}_u$ and $\boldsymbol{\eta}_v$ therefore satisfy a $2N$ linear ODE system and their long-time behaviour can be studied using the eigenvalues of the matrix

$$(7) \quad \begin{pmatrix} f_u \mathbf{I} + \varepsilon \mathbf{L} & f_v \mathbf{I} \\ g_u \mathbf{I} & g_v \mathbf{I} + \varepsilon \sigma \mathbf{L} \end{pmatrix}.$$

Note that in general analytical progress cannot be expected for large matrices of this form. We can, however, apply a similar technique as in Turing's work, where we expand the dynamics on a well chosen orthogonal basis. In the continuum case one uses the orthogonal basis of eigenfunctions of the Laplace operator and in the network case we can instead use the eigenvectors of the graph Laplacian \mathbf{L} . Note that because \mathbf{L} is a real positive semi-definite matrix the eigenbasis for \mathbf{L} is indeed orthogonal and the corresponding eigenvalues are real and non-negative.

Denote by $\{\boldsymbol{\psi}^{(r)}\}$ the set of N eigenvectors of \mathbf{L} with accompanying eigenvalues Λ_r , which lie in the interval $[0, N]$. We assume for simplicity that the eigenvectors are ordered such that the eigenvalues Λ_r are in increasing order. We then expand the perturbations on this basis

$$(8) \quad \boldsymbol{\eta}_u = \sum_{r=1}^N a_r \exp(\lambda_r t) \boldsymbol{\psi}^{(r)}, \quad \boldsymbol{\eta}_v = \sum_{r=1}^N a_r b_r \exp(\lambda_r t) \boldsymbol{\psi}^{(r)},$$

where λ_r now is the growth rate of the r -th eigenvector direction in the perturbation in the reaction-diffusion system (6). Upon substitution of this ansatz we find the following system for the growth rates λ_r

$$(9) \quad \lambda_r \begin{pmatrix} 1 \\ b_r \end{pmatrix} = \begin{pmatrix} f_u + \varepsilon \Lambda_r & f_v \\ g_u & g_v + \varepsilon \sigma \Lambda_r \end{pmatrix} \begin{pmatrix} 1 \\ b_r \end{pmatrix},$$

which leads to two possible growth rates for the r -th eigenvector

$$(10a) \quad \lambda_r = \frac{1}{2} \left(\mathcal{T}_r \pm \sqrt{\mathcal{T}_r^2 - 4\mathcal{D}_r} \right)$$

$$(10b) \quad \mathcal{T}_r = f_u + g_v + \varepsilon(1 + \sigma)\Lambda_r$$

$$(10c) \quad \mathcal{D}_r = \sigma(\varepsilon\Lambda_r)^2 + \varepsilon\Lambda_r(\sigma f_u + g_v) + f_u g_v - f_v g_u.$$

In order for the steady state $(\mathbf{u}^*, \mathbf{v}^*)$ to become unstable we need to have at least one eigenvector direction for which $\Re(\lambda_r) > 0$ holds. Note that (10) yields two possible solutions for λ_r , but since we are only interested in growth rates with $\Re(\lambda_r) > 0$, we focus attention on the solution with the positive branch. To get an unstable state we find from (10) that we need $\mathcal{D}_r < 0$ for some $r = 1, \dots, N$. Using the fact that \mathcal{D}_r is a quadratic in $\varepsilon\Lambda_r$ the condition $\Re(\lambda_r) > 0$ yields the last two Turing conditions

- iii) $\sigma f_u + g_v > 0$,
- iv) $-(\sigma f_u + g_v)^2 + 4\sigma(f_u g_v - f_v g_u) > 0$,

which is exactly the same as derived in the continuous medium case [15, Chapter 2]. Again this is due to the direct correspondence between the eigenvectors and eigenvalues of the graph Laplacian and their counterpart eigenfunctions and eigenvalues for the Laplacian operator.

The Turing conditions i)-iv) now define a region in the model parameter space (parameters for f and g plus ε and σ) for which the Turing instability as described before is possible. We will discuss this further in Section 3. Note that the preceding analysis is similar to a master stability function type approach, see for example [18, Chapter 18].

Mimura-Murray model. In the remainder of this paper we will make a specific choice for the activator-inhibitor dynamics, namely f and g will follow the predator-prey Mimura-Murray model [13]. The analysis and conclusions previously (and much of what follows), however, are not dependent on a specific choice of model dynamics.

In the Mimura-Murray model we have four parameters a, b, c and d and the functions

$$(11a) \quad f(u, v) = u \left(\frac{a + bu - u^2}{c} - v \right)$$

$$(11b) \quad g(u, v) = v(u - (1 + dv)).$$

Note that this model allows for two positive steady states, i.e. roots to f and g simultaneously. The trivial state $u^* = v^* = 0$ and

$$(12) \quad u^* = -\frac{c}{2} \left(-\left(\frac{b}{c} - \frac{1}{d}\right) - \sqrt{\left(\frac{b}{c} - \frac{1}{d}\right)^2 + \frac{4}{c} \left(\frac{a}{c} + \frac{1}{d}\right)} \right), \quad v^* = \frac{u^* - 1}{d}.$$

With the parameter choice $a = 35, b = 16, c = 9$ and $d = 2/5$ as in [13] this turns into $(u^*, v^*) = (5, 10)$ and one can verify that this satisfies the first two Turing conditions.

Network models. In this paper we consider 4 different small networks. The machinery applies to large networks as well, but the computational costs for section 4 scale badly with the network size and so we only consider graphs with $N = 20$ nodes. We consider the complete graph K_{20} , which has a connection between every node. Furthermore we look at two graphs which follow from a patterning of two-dimensional space, namely with a rectangular or hexagonal grid. These lead to a rectangular lattice graph and a triangular lattice graph. Lastly we consider a class of random graphs which in the limit of $N \rightarrow \infty$ yield scale-free networks, graphs from the Barabási-Albert model [6]. Being inherently a random graph model, we show results for one specific realisation of the model, but with representative behaviour. We choose to initiate the Barabási-Albert model graph with a complete graph of size m and then built the graph using preferential attachment with the same m , which in this paper is taken as $m = 2$. The effect of the initial core of the graph is not immediately clear and we make a few comments in appendix B.

3. PATTERN SELECTION

Having fixed the reaction dynamics the model still has two tunable parameters ε and σ , both related to the diffusion. This is the part where the network topology starts to play a role via the spectrum of the graph Laplacian \mathbf{L} and therefore this is where the results on networks can start to deviate from the continuous media case.

Note that the spectrum of \mathbf{L} is discrete and we can determine for each eigenvector $\psi^{(r)}$ when it has a growth factor with $\Re(\lambda_r) > 0$. To determine the bounding curves in (ε, σ) -space where this happens we look at the marginal stability curves of each of the modes, i.e. the curves in (ε, σ) -space for which $\Re(\lambda_r) = 0$. The union of these curves will define a subsection of the parameter space for which Turing patterns can exist, also known as Turing space [16].

Turing space. Carrying out the analysis of the marginal stability curves (this can be done by solving $\mathcal{D}_r = 0$) we find for each mode the bounding curve

$$(13) \quad \varepsilon_r(\sigma) = \frac{1}{2\sigma\Lambda_r} \left(-(\sigma f_u + g_v) \pm \sqrt{(g_v + \sigma f_u)^2 - 4\sigma(f_u g_v - f_v g_u)} \right).$$

Noting that because this curve has to be real valued we get a condition on σ stating a critical value below which no patterns can form

$$(14) \quad \sigma_c = \frac{1}{f_u^2} \left(f_u g_v - 2f_v g_u + 2\sqrt{f_v g_u (f_v g_u - f_u g_v)} \right).$$

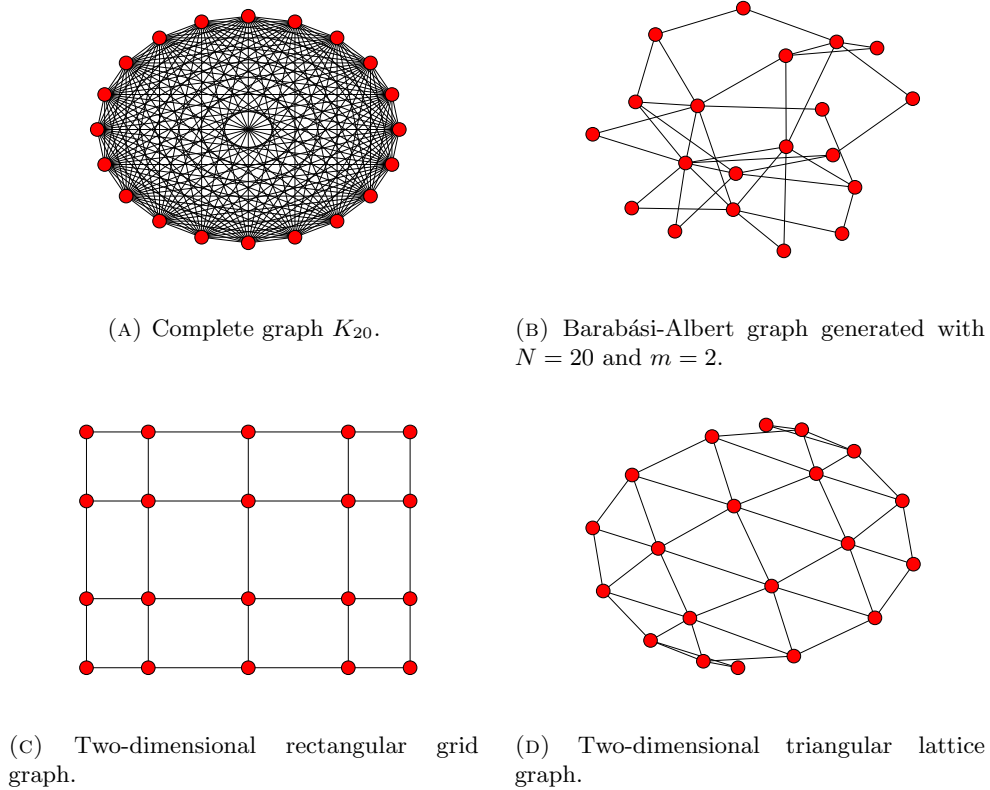


FIGURE 1. Network topologies used in this paper.

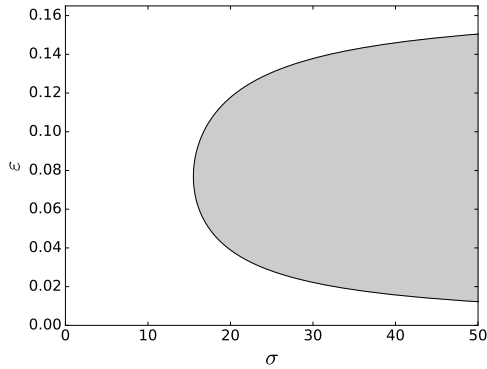
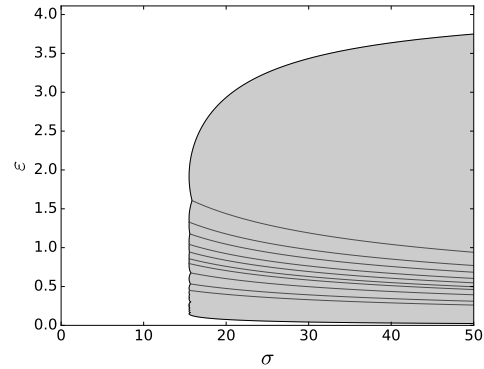
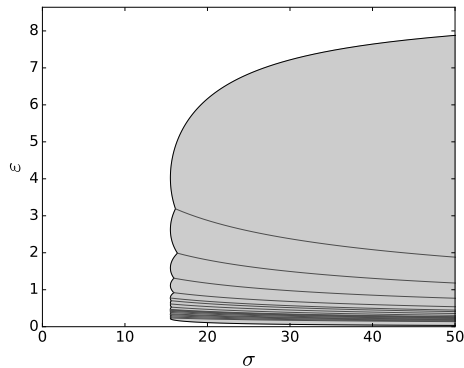
This does not mean that for all (ε, σ) combinations with $\sigma > \sigma_c$ a Turing pattern can form as we will see in Figure 2.

In Figure 2 we have plotted the Turing space for the different networks mentioned in the previous section. We see in grey the union of the regions for which different eigenvectors become unstable.

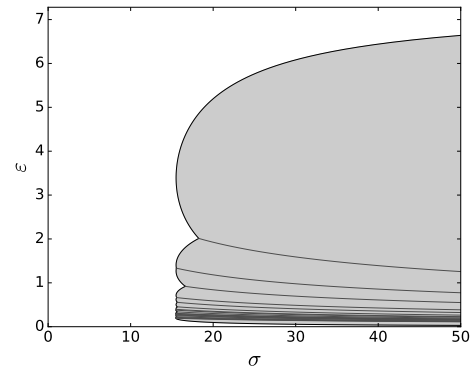
From these graphs we can observe that in general the Turing space will be larger if the network admits various eigenvectors with different eigenvalues. In the case of the complete graph with N nodes K_N the eigenvalues are 0 and N with the latter having algebraic multiplicity of $N - 1$. The Turing space therefore is formed purely by one bounding curve and as a result is fairly small, see Figure 2a. In the case of a more diverse spectrum we see that the Turing space, as a combination of the Turing pattern regions with different Λ_r , now forms a larger whole.

A first approximation to the form of the Turing space can be given by considering the bounding curves $\varepsilon_2(\sigma)$ and $\varepsilon_N(\sigma)$, the curves for the smallest and largest non-zero eigenvalues respectively. If connected by the line $\sigma = \sigma_c$ this gives a rough idea of what the Turing space looks like, apart from irregularities which mainly appear around the critical value of σ_c . In the case of the complete graph these two extremal bounding curves are equal, whereas for the other networks they clearly differ. If one wants a Turing space to stretch out for large values of ε , i.e. strong diffusion, one needs consequently to have a network with a small Λ_2 . This does have an intuitive explanation as this smallest non-zero eigenvalue, Λ_2 , the Fiedler eigenvalue, is related to the algebraic connectivity of the network. A small Fiedler eigenvalue will generally mean that the graph can be well partitioned in two components with little connections between them, whereas a larger Fiedler eigenvalue will mean that such a partitioning is harder to make. If we cannot make such a partitioning that will

mean that the network is well connected and if we would combine that with a large diffusion value ε this would intuitively yield a homogeneous state, because the diffusion will quickly even out any local differences. On the other hand with a small Fiedler eigenvalue and a good partitioning of the network one can have the situation where the fast diffusion yields a fairly homogeneous state within the partitions, which are well connected within itself, but not necessarily overall a homogeneous state. This could be because in this case the lack of strong connections between the two partitions counterbalances the high diffusion constant.


 (A) Complete graph K_{20} .

 (B) Barabási-Albert graph generated with $N = 20$ and $m = 2$.


(C) Two-dimensional rectangular grid graph.



(D) Two-dimensional triangular lattice graph.

FIGURE 2. Turing space for different network topologies as shown in Figure 1. Grey area shows the part of space for which Turing patterning is possible based on analysis in section 2. Dark grey lines within the Turing space show the bounding curves for regions with different fastest growing eigenvectors.

Mode selection. From the growth rates one could in theory predict in which eigenvector direction the perturbation will grow initially, namely one might expect that the eigenvector with the largest growth rate λ_r will dominate the initial growth phase. From this we can derive another set of bounding curves, which divide Turing space into regions in which one of the eigenvectors will dominate, something first done in [1] for the continuum case. This can be achieved by setting the growth rate of two adjacent eigenvectors equal to each other, i.e. find the curves in (ε, σ) -space for which $\lambda_r = \lambda_{r+1}$ for $r = 1, \dots, N - 1$. This prediction of what the pattern could look like is known

as mode selection. These dividing curves are also shown in Figure 2 as dark grey lines. Note that these regions are ordered top to bottom from the smallest to the largest non-zero eigenvalue, i.e. the uppermost region corresponds to the part of space in which the eigenvector $\psi^{(2)}$ dominates.

Mode selection could be expected to work well in cases where only one of the eigenvectors will have $\Re(\lambda_r) > 0$. One can, however, ask whether this still holds true if we have multiple eigenvectors having positive growth rates. Another issue is that the above analysis is based on a linear analysis, whereas the final pattern established will follow the non-linear dynamics from (5). In the continuum case this non-linearity can cause the excitation of higher order harmonics [15, Chapter 2], but it is not clear whether such effects will still hold true in the network case. Nakao and Mikhailov already mentioned that the mode selection in networks differs significantly from the continuum case [17]. They found that the final pattern does not correspond to what they call the critical eigenvector, i.e. the eigenvector with the largest growth rate.

We now show, at least empirically, that this depends on where in the Turing space one looks. First we fix $\sigma = 25$, but note that this is completely arbitrary as long as $\sigma > \sigma_c$. We take the Barabási-Albert graph as example and look at the decomposition of the solution \mathbf{u} at time $t = 1000$ started from random initial conditions for different values of ε . We decompose the vector $\mathbf{u} - \mathbf{u}^*$ onto the eigenbasis to see which of the eigenvectors is most strongly present in the established Turing pattern. In Figure 3 we see that it is mainly in the region $\varepsilon < 1$, where many modes have $\Re(\lambda_r) > 0$, that we do not see a clear eigenvector direction being favoured. In more detail, it looks as if a strong mode selection is present for the first few eigendirections and high values of ε , whereas for lower values of ε and thus eigenvectors with higher eigenvalues it seems as if the final Turing pattern consists more of a mixture of different eigenvectors. This could explain why Nakao and Mikhailov observed that final time Turing patterns on networks deviate strongly from the critical eigenvector directions and therefore behave very different from the continuum case [17]. The parameter values chosen in their study might have focussed on a subset of the Turing space where the different eigenvector directions mix such as in Figure 3 for $\varepsilon < 1$. It would be interesting to see whether their conclusions still hold if a different value for ε is chosen.

4. ROBUSTNESS OF TURING PATTERNS

There are many ways to study the ‘robustness’ of Turing patterns. Research effort has focussed on the classical setting of reaction-diffusion PDEs and people have looked at, amongst others, the effect of external noise and growth on the type of pattern that develops [12], the effect of boundary and initial conditions [1] and the effect that noise on the model parameters has on the mode selection [16]. Such robustness questions have to our knowledge not been addressed in the networks case. Again, there are many ways one could set out to study this problem, the aforementioned ones or the effect growth or shrinking of a network for example. In this paper, however, we look at a type of robustness that has been studied previously, though in a different context, in the networks literature, namely the network resilience to the process of removing some fraction of the graph connections, also known as (bond) percolation.

Percolation. In this paper we consider the case of bond percolation, i.e. the process where edges between nodes can get removed from the graph, as opposed to site percolation which is the process that removes nodes (and its accompanying edges) from the graph. In most networks studies one is interested in the size of a giant component in the network. For this study, however, we will look at ‘the size of a Turing pattern’, something that has to be defined in more detail later.

For simplicity we only consider percolation process with uniform removal of edges, non-uniform removal being an interesting future direction. In this model we define an occupation probability ϕ which will parametrise the percolation process. This denotes a probability for each edge that it is present in the percolation network, i.e. when $\phi = 0$ none of the edges are present whereas for $\phi = 1$ all of them are present. Given an initial network, such as in Figure 1, one can now ask what happens when ϕ varies between 0 and 1.

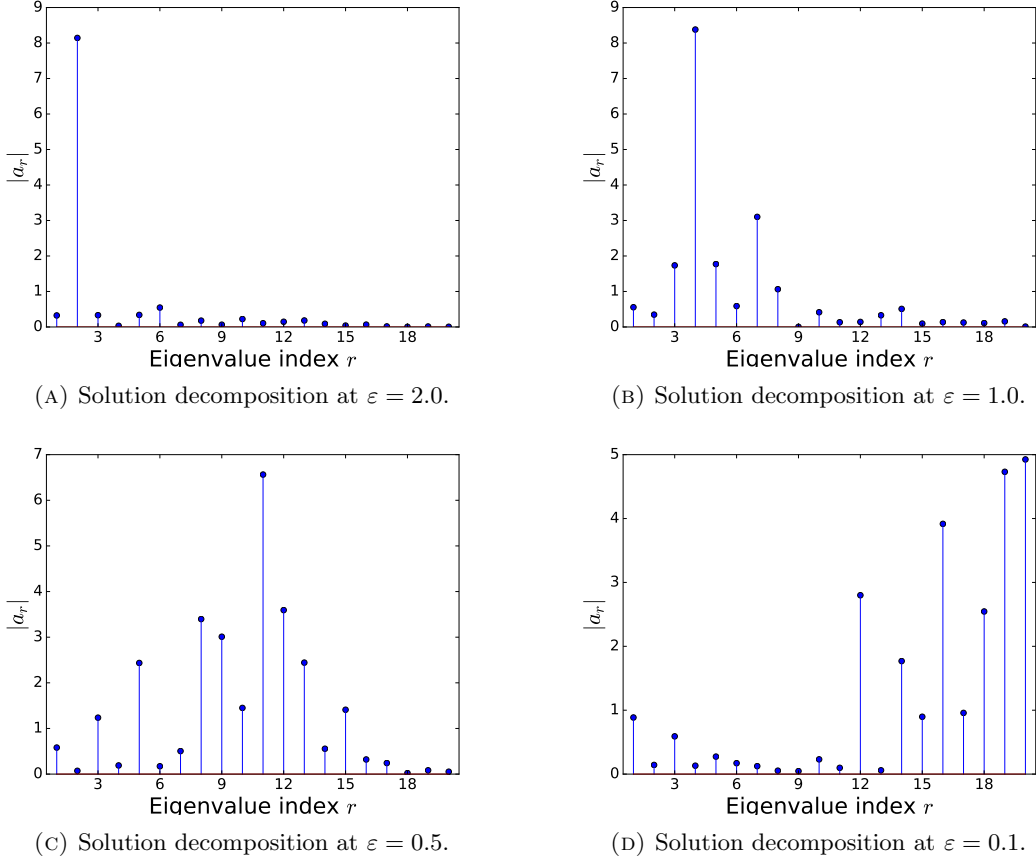


FIGURE 3. The eigendecomposition of the solution for the Barabási-Albert graph at $\sigma = 25$ and time $t = 1000$ for various ε within the Turing space. The strength of eigenvector $\psi^{(r)}$ in the solution is given by $|a_r|$ and the eigenvalues are ordered.

Unfortunately analytical calculations are not tractable in this case so we have to resort to numerical simulations. Suppose that we want to measure some quantity $Q(\phi)$ on the network, we could then fix our ϕ , generate networks with that ϕ and calculate the quantity Q . This does, however, become arduous if we want to calculate Q whilst varying ϕ continuously between 0 and 1. Therefore we use a trick, common in the algorithmic study of percolation, where we define a similar quantity function $Q(m)$ which now denotes Q in a network with m out of a total of M edges present. Note that in a percolation process with occupation probability ϕ , the probability to observe a network with m out of M edges is given by the binomial distribution, i.e. this probability is equal to $\binom{M}{m}\phi^m(1-\phi)^{M-m}$. This allows us then to average over all the possible networks in a percolation process to write

$$(15) \quad Q(\phi) = \sum_{m=0}^M \binom{M}{m} \phi^m (1-\phi)^{M-m} Q(m).$$

This now means that if we find $Q(m)$ for a finite set of values for $m = 0, \dots, M$ we can calculate Q for all values of ϕ . Note that there are many networks possible for a given value of ϕ or m and therefore we apply a Monte Carlo type approach where we find an expected value of Q for given ϕ or m by generating many sample networks and calculating the average value of Q over

these. A simple way to generate a sample graph with m edges given a graph of $m + 1$ edges is to simply remove one edge, rather than generating a network from scratch. Therefore a strategy to efficiently do this is by starting with a graph with M edges and then one by one remove an edge and calculate $Q(m)$ until no edges are remaining. This then has to be repeated many times to get the estimate of Q .

Numerical results. As mentioned before we use a Monte Carlo approach to calculate Turing pattern robustness on the four different networks depicted in Figure 1. We therefore calculate 15000 samples for each value of m , meaning that we effectively run $15000M$ simulations of a system of 40 ODEs. From this we can see why this approach scales badly with the network size, because the number of edges increases roughly quadratic with network size and also the time to compute the ODE solution will scale at least linearly with network size.

To look at the size of a Turing pattern the long-time ($t = 1000$) solution to the system of ODEs (5) using the *LSODA* integration procedure, which is a robust method for problems of this kind where we observe a mixture of non-stiff reaction and stiff diffusion behaviour. Given such a long-time solution (\mathbf{u}, \mathbf{v}) we then calculate the deviation from the homogeneous steady state by considering the vector $(\mathbf{u}, \mathbf{v}) - (\mathbf{u}^*, \mathbf{v}^*)$ and calculate its magnitude. There are of course different ways to define magnitude for a vector but here we consider the l_1, l_2 and l_∞ norms.

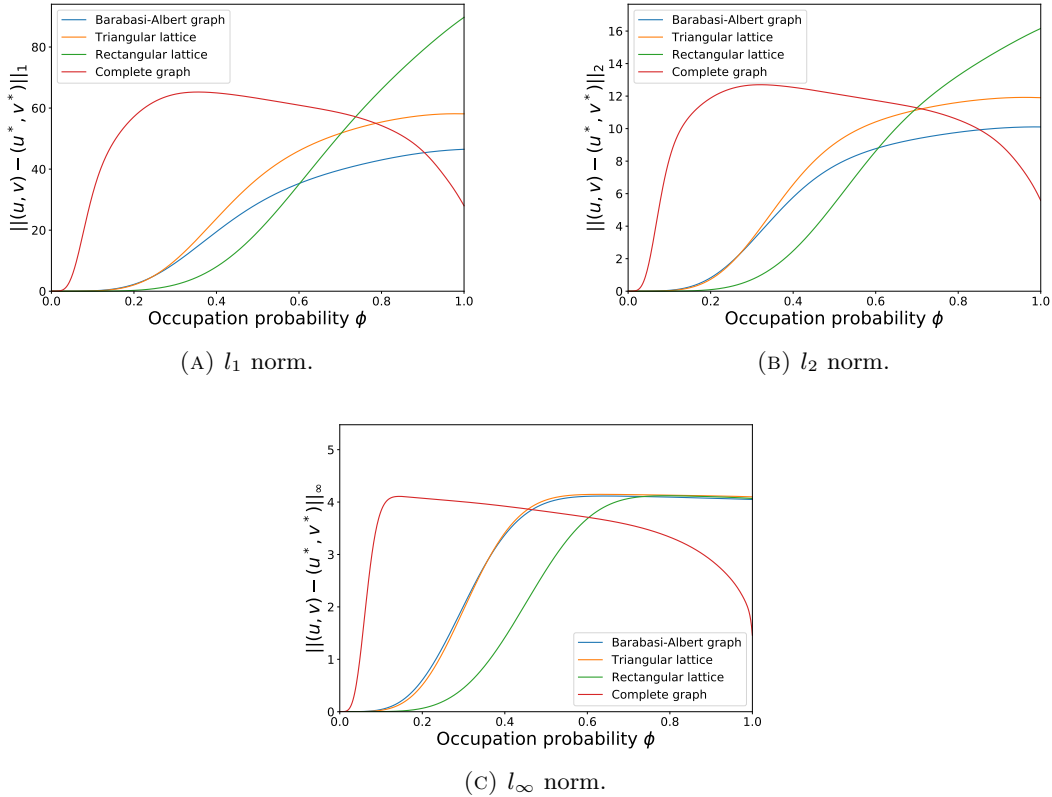


FIGURE 4. Different norms of Turing pattern for $\sigma = 25$ and $\varepsilon = 0.12$ at $t = 1000$ under a bond percolation process as a function of the bond occupation probability ϕ .

The result of a numerical simulation for $\varepsilon = 0.12$ is shown in Figure 4. We first note that having equal norm does not necessarily mean that two patterns are equal, the norms can only be used to make statements about the magnitude of either the whole pattern or just the maximal value. So even though pattern norm stays roughly equal for moderate values of ϕ in the case of the Barabási-Albert graph and the triangular lattice, it might well be the case that the actual form of the pattern changes. Part of the difference in behaviour under the percolation process can be explained by the fact that the different graphs have different number of edges. One might expect that if a graph has more edges it can be more robust to percolation. This can not serve as a full explanation, however, as the Barabási-Albert graph and triangular lattice show similar behaviour even though they have different connectivity and number of edges.

In Figure 5 we plot the distribution of Turing pattern l_2 -norms as seen under the percolation and this shows the non-uniqueness of Turing patterns. We see a discrete band pattern when the occupational probability decreases, for which we can not offer an explanation currently.

We also see that for the complete graph there is an initial rise in the size of the Turing pattern before the pattern gets destroyed for low occupational probability. This effect is even more pronounced if we repeat the experiment for higher value of diffusion, $\varepsilon = 1.00$, as shown in Figure 6. Here we are initially not in the Turing space, so Turing patterns can not form. If edges get removed to create a sparser topology this stimulates the creation of Turing patterns. This can be understood by considering the Fiedler eigenvalue under this percolation process. By removing nodes the Fiedler eigenvalue will decrease, it gets easier to partition the graph, and therefore the Turing space grows such that the point $\varepsilon = 1.00$ moves into the Turing space.

5. DISCUSSION

In this paper we looked at the extension of Turing's classical pattern formation analysis to different network topologies. Many analogies form between the case of a continuous medium and that of the network in terms of the Turing conditions and consequently the allowed parameters that yield self-organising patterns. There are, however, differences between the classical framework and the networks framework, because of the extra ingredient of network topology. We showed in Section 3 that even though networks can have an equal number of nodes their Turing space can vastly differ. The Turing space is fully determined by the graph Laplacian spectrum and therefore intimately linked to the network connectivity. It was mentioned in [17] that mode selection, i.e. the fact that Turing patterns can be decomposed on only a small number of eigenvector directions, was not present in the case of networks. We did, however, show empirically that this depends on the model parameters chosen and thus the location in Turing space.

Having a network topology raises the question how robust the generation of Turing patterns is with respect to changes in this topology. This is studied in Section 4 by applying a bond percolation process to the graphs studied in the paper. We performed an initial calculation for four networks, with different topologies and the effect of percolation. We saw that percolation can not just decrease the size of Turing patterns, but also enables the creation of Turing patterns in certain parts of Turing space. As one can expect we see different behaviour depending on the topology and a detailed study of the results could be done in future work.

Lastly we mention a few possible extensions of the work in this paper. First would be the effect of site percolation rather than bond percolation. This should be computationally less intensive as it scales with the number of nodes N , rather than the number of edges, which is roughly $\mathcal{O}(N^2)$. The application of the methods in this paper to larger networks would be interesting as well, because many network models have properties that only hold in the limit $N \rightarrow \infty$ and therefore there might be a difference in behaviour for networks with large N (compared to $N = 20$ as done in this paper). Lastly we mention again the effect of growth on Turing patterns. In the Barabási-Albert model, for example, one can construct a chain of networks, increasing at every step. One could see whether the patterns observed during this process change shape and size. Furthermore this adds

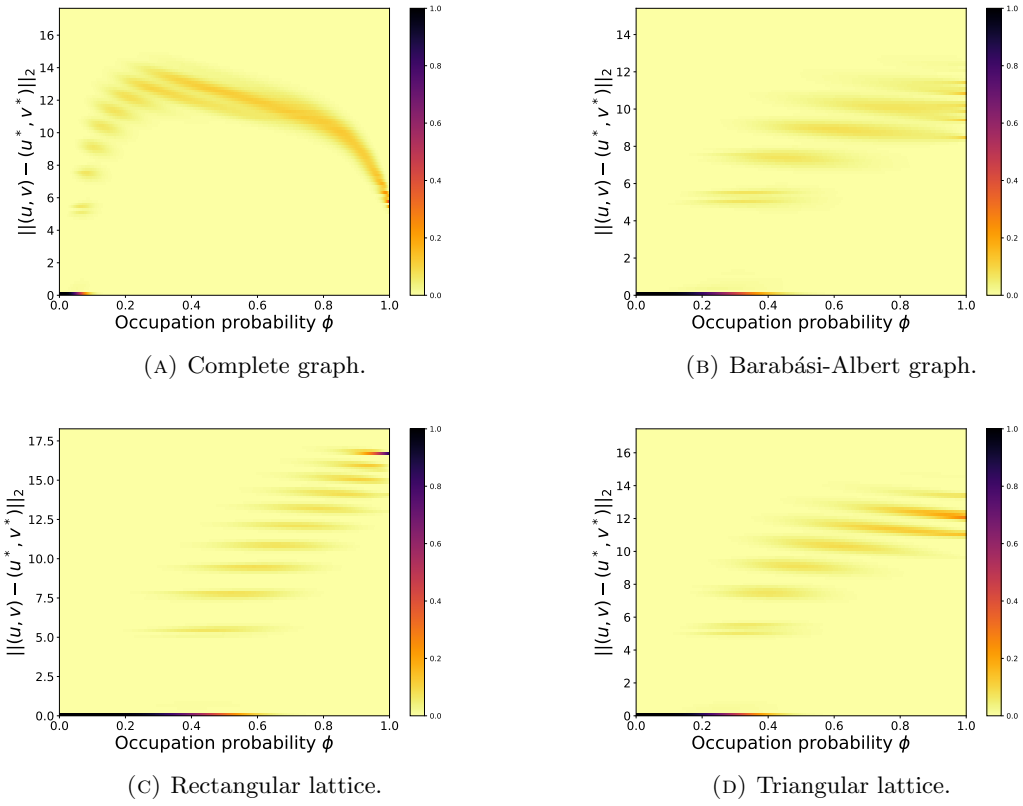


FIGURE 5. Distribution of the l_2 -norm under percolation as observed by Monte Carlo simulation. Results show discrete bands of allowed Turing patterns for all graphs.

another timescale to the problem, namely that of the network growth, it would be interesting to see whether this effects the outcome as well.

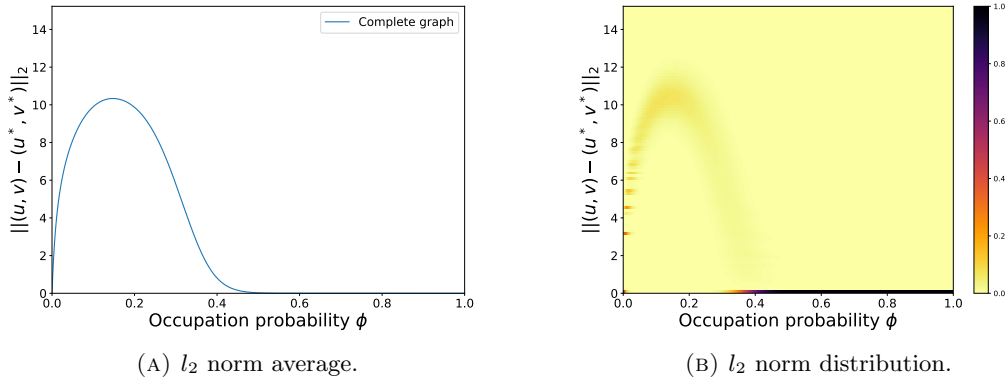


FIGURE 6. l_2 norm of Turing pattern for $\sigma = 25$ and $\varepsilon = 1.00$ at $t = 1000$ under a bond percolation process as a function of the bond occupation probability ϕ on the complete graph K_{20} . *Left*: average distribution of the l_2 norm. *Right*: empirical distribution of the l_2 norm. This distribution displays a discrete character for low values of occupational probability ϕ .

APPENDIX A. CODE

All numerical results and figures in this paper are generated using Python and the networks package *NetworkX*. Python code for much of the work in this paper and to generate the figures will (soon) be available for personal use from

<https://bitbucket.org/CasperBeentjes/turing-patterns-on-networks>.

APPENDIX B. BARABÁSI-ALBERT MODEL INITIATION

As mentioned in section 2 it is not specified by the model itself what the initial core of nodes should be to start the model with. Suppose we want to generate a Barabási-Albert graph with N nodes starting from an initial core network of m_0 nodes and adding a new node at each step with m connections to the previously existing nodes using preferential attachment. For simplicity we consider the case that $m_0 = m$, which is the choice made by the Python networks package *NetworkX*. Of main interest in this paper is the Laplacian spectrum of the resulting graph and in Figure 7 we see the effect of the different initial configurations on the spectrum. The difference in behaviour can be understood from considering what happens when $m \rightarrow N$. In the case of initialising with a complete graph the Barabási-Albert graph will be very similar to a complete graph and therefore have a Fiedler eigenvalue close to N . In the case of the initialisation of an empty graph we see that in the limit $m \rightarrow N$ the graph becomes very similar to a star graph, which has a known Fiedler eigenvalue of 1 for any size. The behaviour of the largest eigenvalue seems to be fairly independent of the choice of initial configuration, which can be understood from the bounds on it which only involve the maximal degree of nodes in the network [18, Chapter 18].

The effects of the initial configuration in the Barabási-Albert model has been studied before using random initial conditions [8], but this mainly focussed on the power-law behaviour of the resulting networks. If one were to use a growing Barabási-Albert model to study the influence of growth on the Turing patterns one should make sure that the initial core does not effect the results, which is most likely achieved by taking an initial core which is much smaller in size than the final network size. That way in the limit of many nodes being added the effect of any initial configuration becomes negligible.

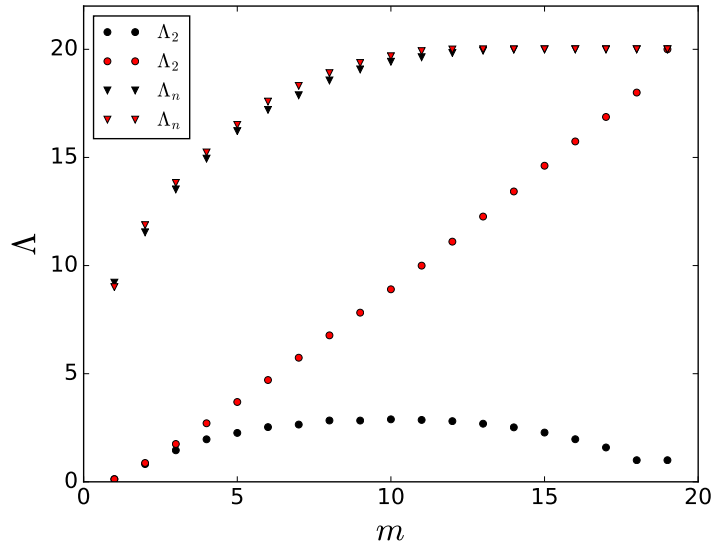


FIGURE 7. The (average) behaviour of the smallest and largest non-zero eigenvalues of the Laplacian for a Barabási-Albert model graph as a function of m for $N = 20$, note that $1 \leq m < N$. The different initial configurations are depicted in different colours, red for an initial complete graph of size m and black for an initial unconnected graph of size m .

REFERENCES

1. Arcuri, P. & Murray, J. D. Pattern sensitivity to boundary and initial conditions in reaction-diffusion models. *Journal of mathematical biology* **24**, 141–65 (1986).
2. Asllani, M., Busiello, D. M., Carletti, T., Fanelli, D. & Planchon, G. Turing patterns in multiplex networks. *Physical Review E* **90**, 042814 (2014).
3. Asllani, M., Carletti, T. & Fanelli, D. Tune the topology to create or destroy patterns. *European Physical Journal B* **89** (2016).
4. Asllani, M., Challenger, J. D., Pavone, F. S., Sacconi, L. & Fanelli, D. The theory of pattern formation on directed networks. *Nature communications* **5**, 4517 (2014).
5. Asllani, M., Di Patti, F. & Fanelli, D. Stochastic Turing patterns on a network. *Physical Review E* **86**, 046105 (2012).
6. Barabási, A.-L. & Albert, R. Emergence of Scaling in Random Networks. *Science* **286**, 509–512 (1999).
7. Getling, A. V. *Rayleigh-Bénard convection: structures and dynamics* (World Scientific, 1998).
8. Guimarães, P. R. *et al.* Random initial condition in small Barabasi-Albert networks and deviations from the scale-free behavior. *Physical Review E* **71**, 037101 (2005).
9. Hanski, I. Metapopulation dynamics. *Nature* **396**, 41–49 (1998).
10. Horsthemke, W., Lam, K. & Moore, P. K. Network topology and Turing instabilities in small arrays of diffusively coupled reactors. *Physics Letters A* **328**, 444–451 (2004).
11. Kouvaris, N. E., Hata, S. & Guílera, A. D. Pattern formation in multiplex networks. *Scientific Reports* **5**, 10840 (2015).

12. Maini, P. K., Woolley, T. E., Baker, R. E., Gaffney, E. A. & Lee, S. S. Turing's model for biological pattern formation and the robustness problem. *Interface Focus* **2**, 487–496 (2012).
13. Mimura, M. & Murray, J. D. On a diffusive prey-predator model which exhibits patchiness. *Journal of Theoretical Biology* **75**, 249–262 (1978).
14. Moore, P. K. & Horsthemke, W. Localized patterns in homogeneous networks of diffusively coupled reactors. *Physica D: Nonlinear Phenomena* **206**, 121–144 (2005).
15. Murray, J. D. *Mathematical Biology II: Spatial Models and Biomedical Applications* 3th (ed Murray, J. D.) (Springer-Verlag, New York, 2003).
16. Murray, J. D. Parameter space for turing instability in reaction diffusion mechanisms: A comparison of models. *Journal of Theoretical Biology* **98**, 143–163 (1982).
17. Nakao, H. & Mikhailov, A. S. Turing patterns in network-organized activator-inhibitor systems. *Nature Physics* **6**, 544–550 (2010).
18. Newman, M. E. J. *Networks : an introduction* (Oxford University Press, 2010).
19. Othmer, H. G. & Scriven, L. E. Instability and dynamic pattern in cellular networks. *Journal of Theoretical Biology* **32**, 507–537 (1971).
20. Othmer, H. G. & Scriven, L. E. Non-linear aspects of dynamic pattern in cellular networks. *Journal of Theoretical Biology* **43**, 83–112 (1974).
21. Siero, E. *et al.* Striped pattern selection by advective reaction-diffusion systems: Resilience of banded vegetation on slopes. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **25**, 036411 (2015).
22. Sporns, O., Tononi, G. & Kötter, R. The human connectome: A structural description of the human brain. *PLoS Computational Biology* **1**, 0245–0251 (2005).
23. Turing, A. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* **237**, 37–72 (1952).
24. Wolpert, L. Positional information and the spatial pattern of cellular differentiation. *Journal of Theoretical Biology* **25**, 1–47 (1969).
25. Zhang, D., Gyorgyi, L. & Peltier, W. R. Deterministic chaos in the Belousov-Zhabotinsky reaction: Experiments and simulations. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **3**, 723–745 (1993).