

Sample of Numerical Work

Gabriel Nilsson

Contents

1	Percolation on different grids	2
2	Numerical solver for Laplace's Equation	15
3	The Trace-Determinant Plane	24

1 Percolation on different grids

In this project, I will analyze the percolation, and critical percolation probability of different grids. Percolation can more generally be defined to be when some property propagates from one location to another location, often from one side to the other side of a box or a square.

All percolation models I'll describe will consist of a square or box grid of cells, where each cell is active with a probability p , and inactive with a probability $1 - p$. Percolation occurs if there is a non-broken path using only “active” cells that travel from the top of the grid to the bottom.

For example, in a square grid

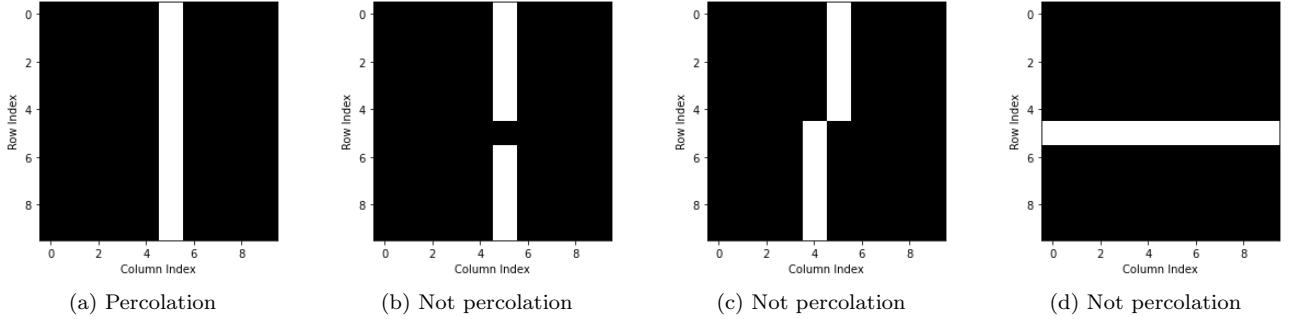


Figure 1: A 10×10 square grid, white cells are “active”. Percolation is only considered from top to bottom, and there has to be an unbroken path of active cells.

Note that I'm only considering two cells to be neighbors if they share a face or a side, not if they only share a node or vertex; hence, Figure 1c is not an example of percolation. This corresponds to using the Von-Neumann neighborhood rather than the Moore neighborhood.

We also have to make a decision as to whether we consider the other boundaries to be periodic or not. For this project I will use periodic boundaries unless stated otherwise.

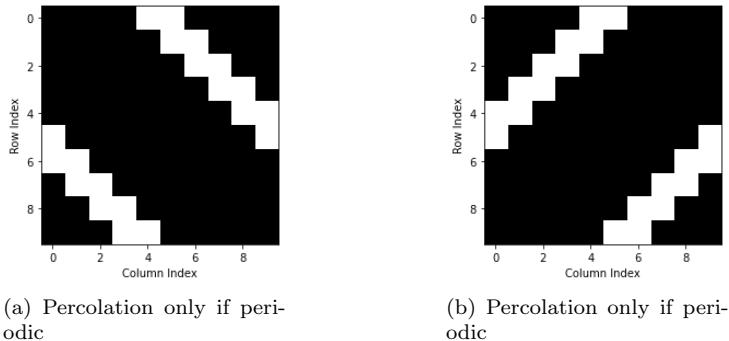


Figure 2: A 10×10 square grid, white cells are “active”. Percolation occurs in both these cases if we let the sides be periodic, but not otherwise.

Now what a simulation run consists of is choosing a type of grid, a size N , and a probability of activation p . Each cell will then randomly and independently be set to active according to p , where we observe whether percolation has occurred.

What is interesting about percolation is that the probability of percolation p_{perc} is not linearly related to the probability p of an individual cell being active, instead we find that percolation almost never occurs when p is less than some critical value p_c , and percolation almost always occurs when p is larger than the critical value p_c . The system undergoes a phase change, where a small change in a parameter creates a sudden and discrete jump in a coarse-grained variable.

We can analyze this phase transition by running the simulation many times for many different values of p , and count up the fraction of simulations that had percolation for that given value of p . For example, in the square case,

One might naively first think that the critical behavior is when p_{perc} surpasses 0.5, although note that this occurs at different points for the different sizes of grids. What we're interested in is the critical point that is

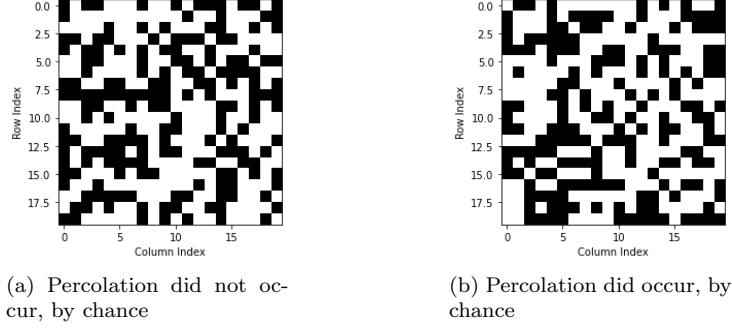


Figure 3: 20×20 square grid, $p = 0.55$ (white is active). Example of how percolation can happen, or not happen, with a given value for p .

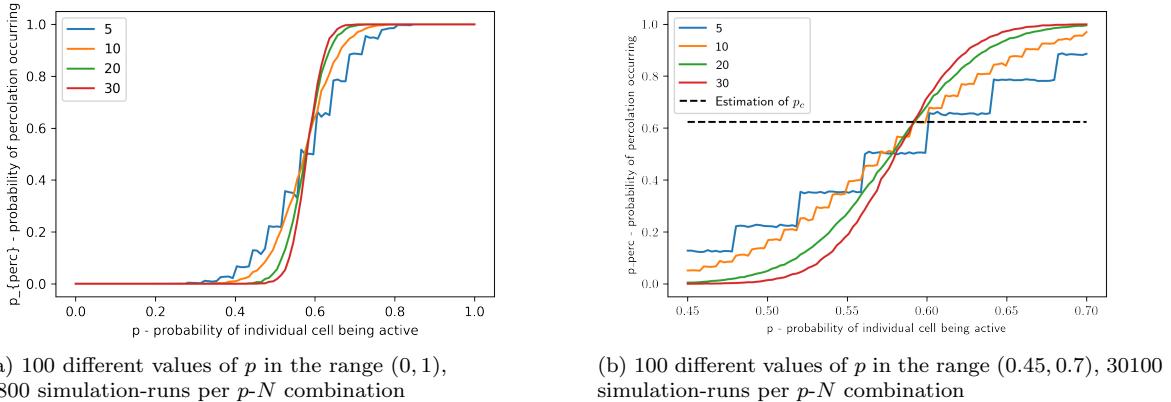


Figure 4: The probability of percolation p_{perc} over the probability of individual cell activation p for square grid, with periodic boundaries. The estimation for the critical probability p_c is highlighted. Each line is a different size of grid, where if $N = 10$ it's a 10×10 grid.

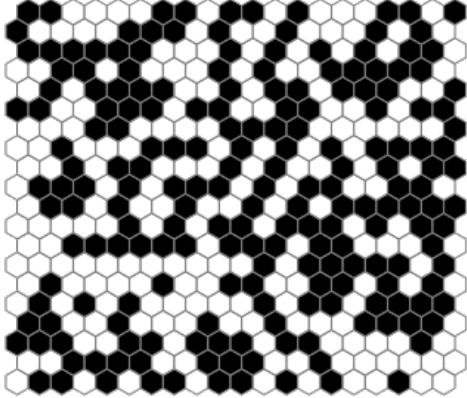
invariant of scale, a point on the graph where p leads to the same p_{perc} , regardless of the size of the grid. The empirical estimation of this point is where the lines for $N = 10$, $N = 20$, and $N = 30$ intersect.

I find that, in a square grid, the critical probability can be estimated to $p_{perc} \approx 0.624$ and $p \approx 0.591$. The true critical probability p_c will satisfy $p_c = p_{perc} = p$ at all scales, and hence I would expect my empirical estimations ($p_{perc} \approx 0.624$ and $p \approx 0.591$) to approach each other as we increase the resolution of p -values.

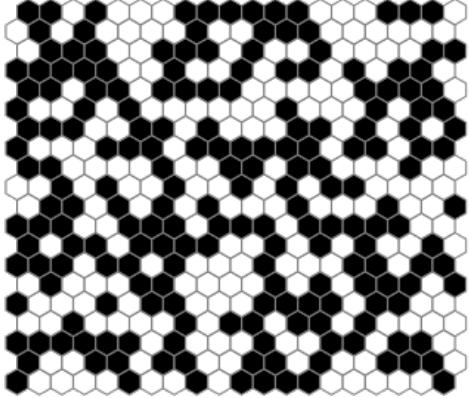
We can explore whether this critical value p_c changes for different types of grids, by performing similar empirical estimations of where the different lines on the p_{perc} plots intersect.

Hexagonal grid

In Figure 5 below, we can see example visualizations of hexagonal grids, one where there is percolation, and one where there is not.



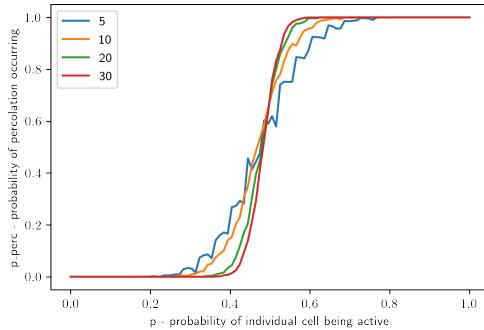
(a) 20×20 hexagonal grid, $p = 0.5$, percolation occurred



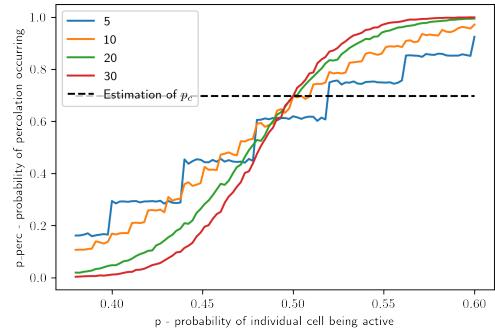
(b) 20×20 hexagonal grid, $p = 0.5$, percolation didn't occur

Figure 5: Example hexagonal grids for $N = 20$, $p = 0.5$. White is active, and periodic boundaries are applied.

We can, just like in the square case, run many simulations and count up the probability of percolation empirically, as a function of cell activation probability p .



(a) 100 different values of p in the range $(0, 1)$,
1100 simulation-runs per p - N combination



(b) 100 different values of p in the range $(0.38, 0.6)$,
6200 simulation-runs per p - N combination

Figure 6: The probability of percolation p_{perc} over the probability of individual cell activation p for a hexagonal grid, with periodic boundaries. The estimation for the critical probability p_c is highlighted. Each line is a different size of grid, where if $N = 10$ it's a 10×10 grid.

Looking at Figure 6, we see that the overall behavior is very similar, in fact the critical probability is estimated to be $p_{perc} \approx 0.633$, compared to 0.624 for the square grid. Although this time there is a larger difference between p_{perc} and p at the point of intersection, with $p \approx 0.495$. Although again, in the theoretical ideal, these values are equal.

Cube grid

We now move into the 3D realm, starting with the cube grid. Everything works the same as in the previous models, except that we now have two different sides which both will act periodically. Note that for the cube grid, we consider cubes that share a face neighbors, not if they only share an edge or a vertex, the 3D case for the von Neumann neighborhood.

We can see an example of a 3D cube grid where $p = 0.2$ in Figure 7.

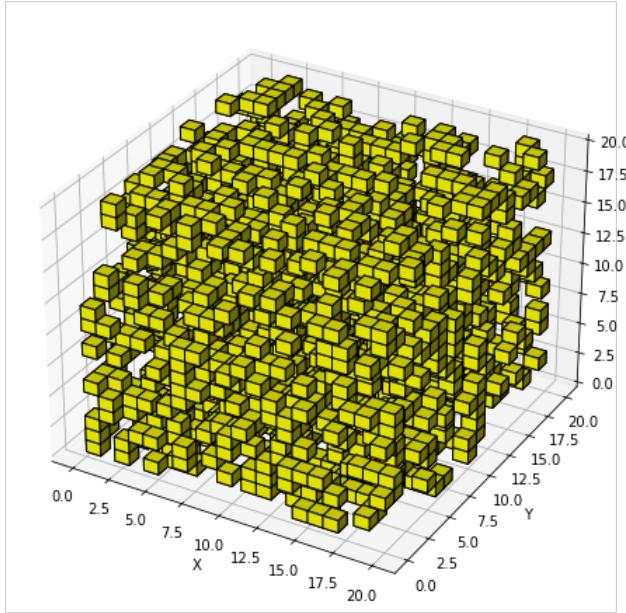


Figure 7: Example of randomly populated 3D grid, $N = 20$, $p = 0.2$, yellow is active.

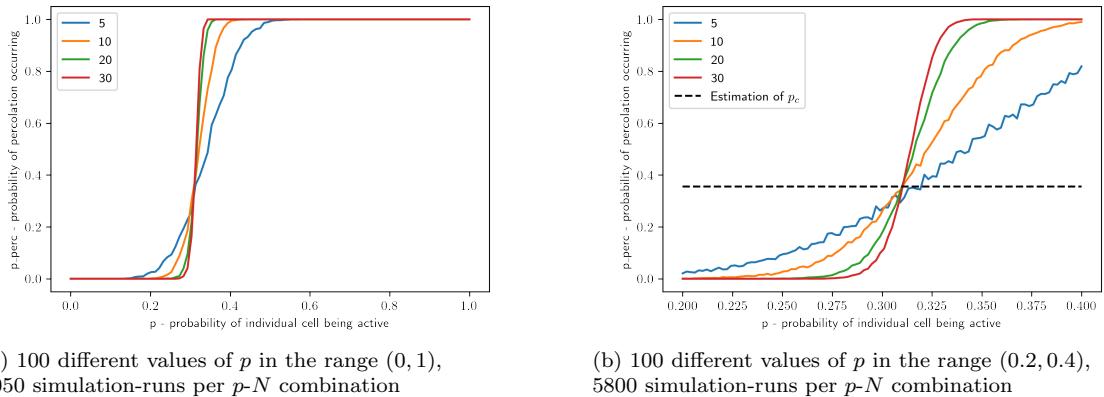


Figure 8: The probability of percolation p_{perc} over the probability of individual cell activation p for a cube grid, with periodic boundaries. The 50% line is highlighted. Each line is a different size of grid, where if $N = 10$ it's a $10 \times 10 \times 10$ grid.

I find that the estimated critical $p_{perc} \approx 0.355$ and that the estimated critical $p \approx 0.311$. These are notably close to each other, which is the expected behavior for perfect renormalization, where the probability of percolation at a single cell p , will equal the probability of percolation at all scales p_{perc} .

RDG-4 Grid

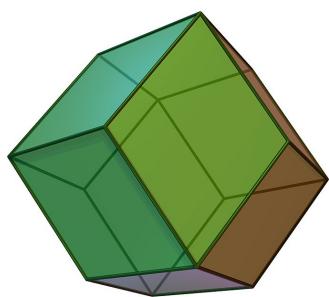
Now we move over to the first non-traditional grid structure. This grid uses rhombic dodecahedra (RD) as cells, a twelve-sided shape that tessellates the volume and hence can create a grid (See Figure 9).

Unlike the square, hexagon, and cube, which all are fully regular polygons and polyhedra, the RD is not fully regular. Specifically, it does satisfy edge- and face-transitivity, but not vertex-transitivity. Fully regular polyhedra satisfy all types of transitivity, and in 3 dimensions, those are the platonic solids. Since only the cube tessellates 3D space, among the platonic solids, we are forced to abandon at least one transitivity property to find another 3D polyhedra that tessellates 3D space.

Transitivity over a component, for example edges, means that we choose some edge, and then translate and rotate the entire polyhedron so that any other edge ends up perfectly where the selected edge was. After the rotation and translation, the entire polyhedra needs to occupy the same space that it occupied before the rotation and translation. In simpler terms, transitivity means that there is only one unique type of that component. A

cube is vertex-transitive because there's no way for me to tell two different vertices apart. Meanwhile for the rhombic dodecahedron, there are two distinct types of vertices. If you look closely at Figure 9a you can see that there is one type of vertex where three edges meet and one type of vertex where four edges meet. While this means there is an inherent asymmetry in the shape, there's also a sense of where the rhombic dodecahedron is as regular as possible without actually being fully regular (a platonic solid), since the only difference is that it has two types of vertices instead of one. As a matter of fact, if you look at the platonic solid analog in 4D (the Regular 4-polytopes), we have 6 instead of the familiar 5 platonic solids. That's because while each 3D platonic solid stays regular as you stretch it into 4D, the rhombic dodecahedra becomes our new platonic solid as we stretch it into 4D (or the 4D analog of a rhombic dodecahedron called the icositetrahedron).

Regardless of this pretty feature, our RD is still asymmetrical, and for this reason we must isolate two distinct orientations of the RD. One orientation where it has acute vertices (4 edges to a vertex) aligned with vertical axis, and one orientation where it has obtuse vertices (3 edges to a vertex) aligned with the vertical axis. The one with acute vertices along the z -axis creates a square profile as you look at it orthographically from the z -axis. Hence, I will label this grid as RDG-4. The one with obtuse vertices along the z -axis creates a hexagon profile as you look at it orthographically from the z -axis. Hence, I will label this grid as RDG-6. These orientations matter because in the RDG-4, each cell has 4 neighbors in the layer above it, 4 on the same layer, and 4 on the layer below. While for RDG-6, each cell has 3 neighbors on the layer above, 6 on the same layer, and 3 on the layer below. It is reasonable to believe that this difference in neighborhood allocation can have a significant impact on the percolation behaviors specifically along the z -axis, which is why I treat them separately.

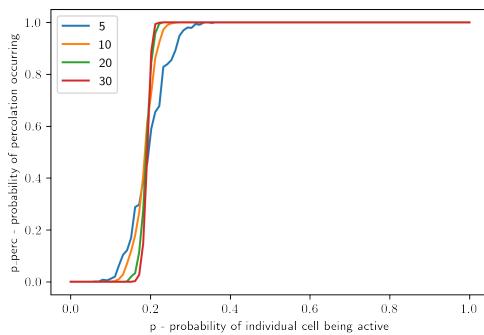


(a) A rhombic dodecahedra (Figure is from [1])

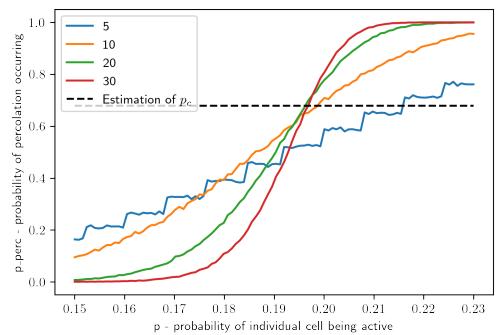


(b) a tessellation of rhombic dodecahedra, or a rhombic dodecahedron grid (RDG) (Figure is from [2])

Figure 9: Visual examples of the rhombic dodecahedron, and a rhombic dodecahedron grid (also known as a rhombic dodecahedron honeycomb)



(a) 100 different values of p in the range $(0, 1)$,
2050 simulation-runs per p - N combination



(b) 100 different values of p in the range $(0.15, 0.23)$,
11500 simulation-runs per p - N combination

Figure 10: The probability of percolation p_{perc} over the probability of individual cell activation p for a RDG-4 grid, with periodic boundaries. The 50% line is highlighted. Each line is a different size of grid, where if $N = 10$ it's a $10 \times 10 \times 10$ grid.

I find that the estimated critical percolation probability is $p_{perc} \approx 0.680$ and the estimated individual cell

activation probability is $p \approx 0.197$. Quite a large difference.

RDG-6 Grid

We now go over to the RDG-6 grid, which is just like RDG-4, except oriented such that the obtuse vertices are aligned in the z -axis. This way each cell has 3 neighbors on the layer above and below, and 6 neighbors on the same layer.

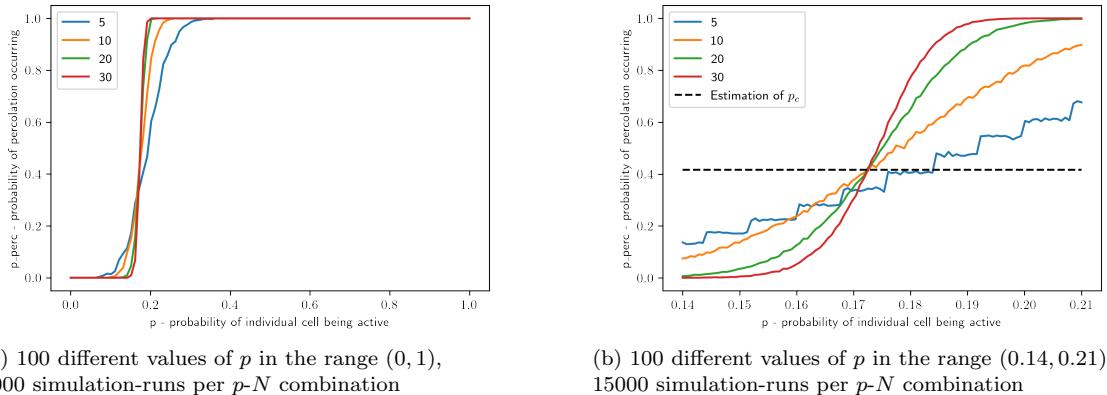


Figure 11: The probability of percolation p_{perc} over the probability of individual cell activation p for a RDG-6 grid, with periodic boundaries. The 50% line is highlighted. Each line is a different size of grid, where if $N = 10$ it's a $10 \times 10 \times 10$ grid.

I find that the estimated critical percolation probability is $p_{perc} \approx 0.417$ and the estimated individual cell activation probability is $p \approx 0.173$. Like RDG4, quite a large difference. This significant difference in both RDG4 and RDG 6 might be related to the non-regularity of the cell shape.

Renormalization

We can make use of renormalization theory to try to predict, or in this case reproduce, the critical probabilities theoretically. Renormalization is about describing change over different scales rather than time, where the interpretation of stationary states is self-similarity over scales, rather than self-similarity over time. In other words, If I find through mean field analysis that an SIR model is at a stationary state, it means that the population will be in the same state at time t_0 , as it will be at some later time t_1 . A stationary state under renormalization means that the system will have the same properties, or be in a similar state, in the scale $L \times L$, as well as if we zoom out to a scale $L' \times L'$. In our case, this property is percolation. For example, the trivial stationary states under renormalization are when $p = 1$ and $p = 0$, at a scale 1×1 we're certain that percolation occurs and doesn't occur respectively, crucially this is the case for any scale $L' \times L'$. If instead $p = 0.1$, there is a 10% chance of percolation at the scale 1×1 , but as we let $L \rightarrow \infty$, the probability of percolation will approach 0. Hence this is not a stationary state under renormalization, since the percolation probability is not stationary over different scales. Look back at the previous plots displaying percolation probability over cell activation probability, and notice that this notion of critical probability corresponds to the point where all lines intersect, as this is a value of p that leads to the same p_{perc} , invariant of scale (at least theoretically).

Square grid

The core process of performing renormalization is to find how our property on one scale relates to the same property at the next scale. If we let p_n represent the probability of percolation for an $n \times n$ square grid, then p_1 is our base case of a single cell (“Single cell level” in Figure 12). We can now calculate exactly how p_1 relates to p_2 , by enumerating all possible microstates in a 2×2 square grid, and sum the probability for each microstate.

We find that percolation occurs in a 2×2 (p_2) if and only if all 4 cells are active (p_1^4), or any three cells are active ($4p_1^3(1 - p_1)$), or if the upper half or lower half is active ($2p_1^2(1 - p_1)^2$). We then have the function Φ relating the base case scale to the next.

$$p_2 = \Phi(p_1) = p_1^4 + 4p_1^3(1 - p_1) + 2p_1^2(1 - p_1)^2. \quad (1)$$

Which is the function displayed in Figure 12.

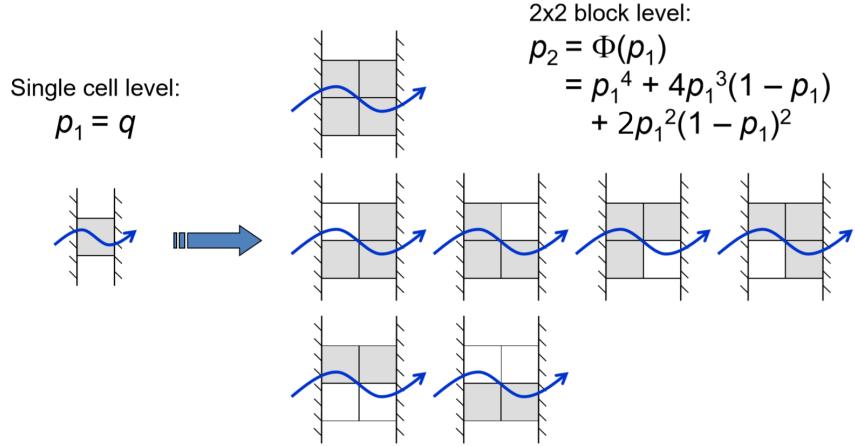


Figure 12: Displays the base case, and calculation for the function Φ (Figure is from [3])

Now we make an inductive step, which builds on a significant assumption. We assume that the equation $p_{2n} = \Phi(p_n)$ holds for all n . As described in Figure 12.

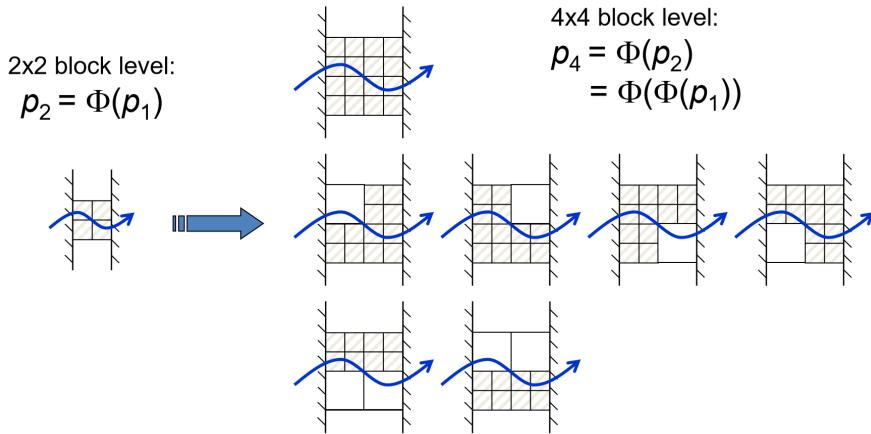


Figure 13: Displays the inductive step from p_2 to p_4 (Figure is from [3])

Here we assume that the relationship between the percolation probabilities between each step of scale are the same. In other words, that the percolation probability of an 8×8 grid depends on the percolation probabilities of its 4 constituent 4×4 quarters, in the exact same way that a 4×4 would depend on its 4 constituent 2×2 quarters. This assumption is not strictly correct, but becomes an increasingly good approximation of reality at larger scales. At small scales, like 5×5 , this error is more significant, hence we see that the 5×5 lines in the percolation probability plots (For example Figure 11) do not intersect the point that is common among all other lines.

We can convince ourselves that this assumption is incorrect by looking at case A and case B in Figure 14. In case A, the bottom right quarter only has a single active cell, and only looking at that 2×2 there's no percolation, hence our function Φ would predict that the overall 4×4 does not percolate since only the bottom-left and top-right are percolating on their own. Case B shows the opposite, here both 2×2 subgrids are percolating on their own, and Φ would predict percolation, but actually it does not percolate. However, as the dimensions become large, the probability of such mismatch cases decreases.

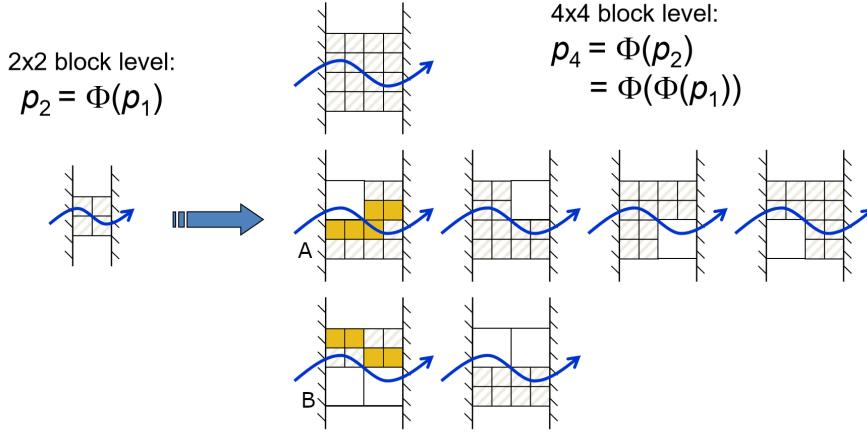


Figure 14: Yellow is active, the figure highlights the invalidity of inductive assumption (Figure is from [3])

Notice that this inductive step effectively is repeated coarse-graining. One can imagine a process where we have a large grid $2^N \times 2^N$, and we iteratively coarse grain by replacing (summarizing “fine-grained” information) each 2×2 by a single cell, which is active if the 2×2 percolates, and inactive if not. We can see this process in Figure 15, where we look at the red 2×2 , and replace it with a single red cell in the next step since the 2×2 percolates from left to right. Each cell in every step is a binary coarse-grained variable that represents whether the underlying 2×2 was percolating or not. If we continue it we eventually get a single cell (rightmost in Figure 15), which now is a coarse-grained binary variable that describes whether the original grid percolates or not. This is what we’re doing when we’re applying the function Φ , and the critical probability p_c is when the probability of individual cells being active is the same across all steps (scales). Although, do remember that this is only an approximation, which is not always accurate.

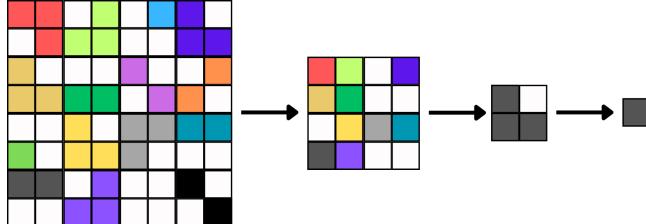


Figure 15: At each step, each 2×2 subgrid is replaced with a single active cell if it percolated from left to right, and an unactive cell if it does not percolate. Colored cells are active, white cells are inactive. This visualizes the iterative logic of the Φ function, and how it’s connected to coarse-graining.

We can now analyze the theoretical expectation of how the percolation probability “evolves” as we increase the scale of our system, or as we “zoom out”. This is well visualized using a cobweb plot (Figure 16). Here we plot the function $p_{2n} = \Phi p_n$ against $p_{2n} = p_n$. Looking at a point on the blue line in Figure 16, we can read off the new probability of percolation on the y -axis. Plugging in the new percolation probability recursively corresponds to “bouncing” against the line function $p_{2n} = p_n$ as the arrows depict, specifically, the arrows show how if we have a percolation probability of 0.5 at some scale n , it will approach a percolation probability of 0 as we increase the scale.

You can also see that there are three stationary points, where the next percolation probability equals the previous one. We can find these points analytically by solving the equation

$$p_n = \Phi(p_n)$$

$$p_n = p_n^4 + 4p_n^3(1 - p_n) + 2p_n^2(1 - p_n)^2$$

where we find the solutions

$$\begin{aligned} p_0^* &= 0 \\ p_1^* &= \frac{\sqrt{5}-1}{2} \\ p_2^* &= 1 \\ p_3^* &= \frac{-\sqrt{5}-1}{2} \end{aligned}$$

Where we discard p_3^* since it's negative and hence outside the range of possible probability values $(0, 1)$. Notice that the nontrivial critical point $p_c = p_1^*$ is the golden ratio conjugate, sometimes called the silver ratio $\frac{\sqrt{5}-1}{2} = \frac{1}{\varphi} = \varphi - 1$, how beautiful.

As described earlier, p_0^* and p_2^* are trivial stationary states, as complete emptiness and complete activation looks identical across all scales. The interesting behavior is at the the critical probability p_1^* , here we have non-trivial self-similarity across all scales. Another way to phrase this phenomena is that if we have an infinite grid at the critical probability, and zoom in on a 20×20 region, it will percolate with the same probability as a 2000×2000 region.

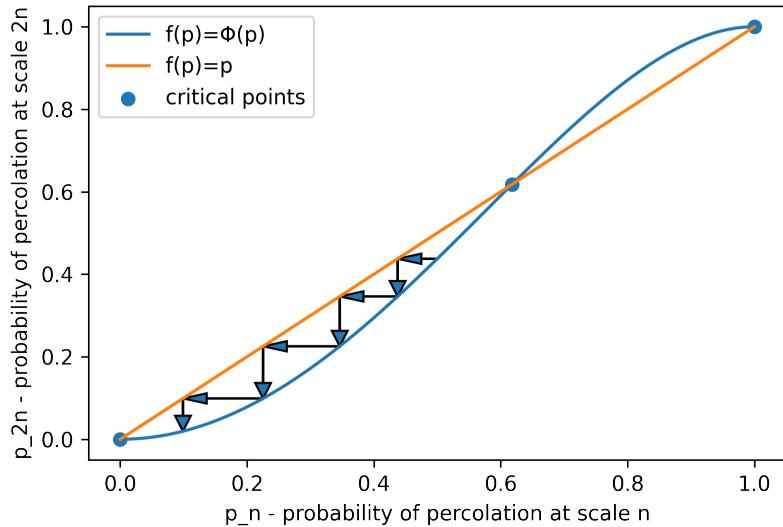


Figure 16: Cobweb plot for renormalization analysis of the square grid. Arrows describe progression of percolation probability according to Φ as we increase scale. Arrows depict evolution of $p_n = 0.5$ as we increase scale.

We can now compare this with the data I collected empirically. We find that it's at this percolation probability that the lines align at, just like estimated earlier. In fact, the relative error between the analytically calculated value, and the empirically estimated one is ($\frac{|\varphi-1-0.624|}{\varphi-1} \approx 0.01$) only about 1%.

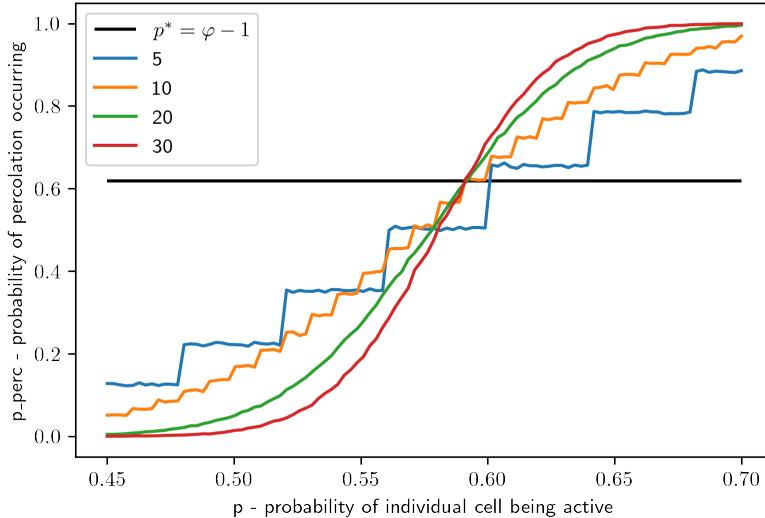


Figure 17: Cobweb plot for renormalization analysis of the square grid. Arrows describe progression of percolation probability according to Φ as we increase scale. Arrows depict evolution of $p_n = 0.5$ as we increase scale.

Hexagon grid

When deriving the function Φ for the hexagonal grid, we have to think about how we relate a cell to a grid “one step” larger, and how that grid relates to a grid “one step” larger, as in Figure 15 for the square grid.

One such mapping, that preserves the hexagonal shape as we scale up is the one found in Figure 18. Although the approach I’ve used for percolation requires two parallel sides, which stay parallel as we increase the scale, so I will opt for the coarsegraining mapping in Figure 19.

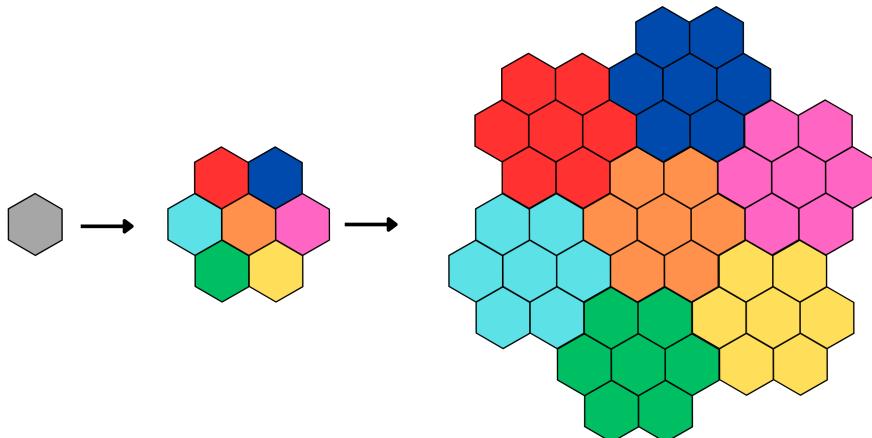


Figure 18: Coarsegraining mapping on hexagonal grid that maintains hexagonal self-similarity on different scales

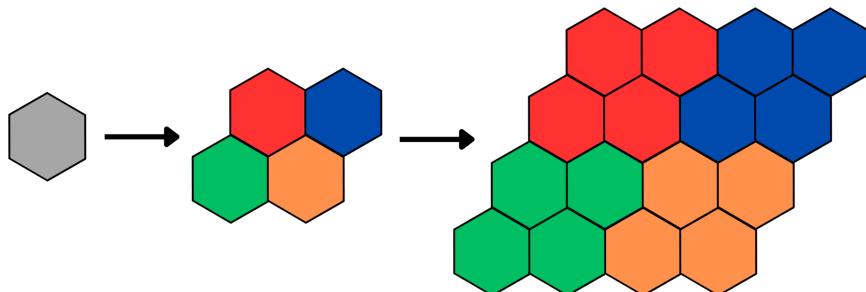


Figure 19: Coarsegraining mapping on hexagonal grid that maintains parallelogram shape on different scales

We can see exactly what the function Φ should look like by considering all combinations of the 2×2 level that percolates (from left to right), as seen in Figure 20.

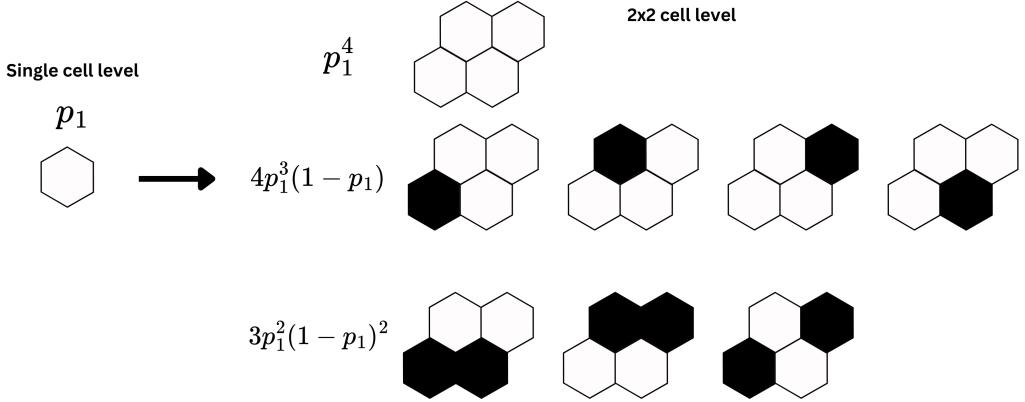


Figure 20: Displays the base case for the hexagonal grid, and calculation for the function Φ . White is activated.

We hence get, for the hexagonal case,

$$p_2 = \Phi(p_1) = p_1^4 + 4p_1^3(1 - p_1) + 3p_1^2(1 - p_1)^2,$$

and with the induction step

$$p_{2n} = \Phi(p_n) = p_n^4 + 4p_n^3(1 - p_n) + 3p_n^2(1 - p_n)^2. \quad (2)$$

Which gives us the following cobweb plot

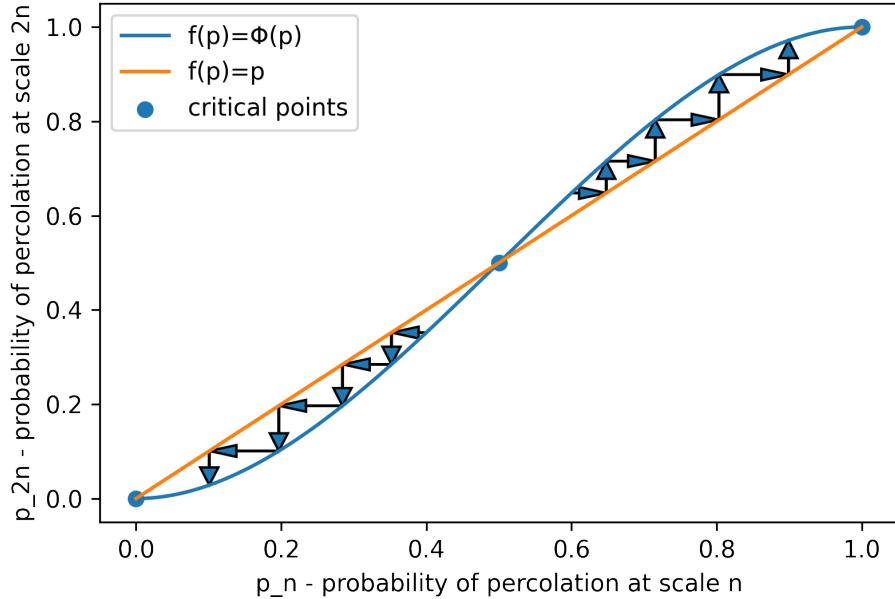


Figure 21: Cobweb plot for the hexagonal grid, arrows highlight divergence when $p_n = 0.4$ and $p_n = 0.6$.

If we solve the equation $\Phi(p_n) = p_n$ to find the critical points we get:

$$p_0^* = 0$$

$$p_1^* = \frac{1}{2}$$

$$p_2^* = 1$$

Where our nontrivial critical point p_1^* is the critical probability of interest p_c . The analytical finding of $p_c = 0.5$ for the hexagonal grid aligns with was found empirically to be the intersection point, but this time along the p -axis rather than the p_{perc} axis. The p_{perc} value at $p = p_c = \frac{1}{2}$ is $p_{perc} \approx 0.697$, which is exactly the site percolation threshold found by [4–6].

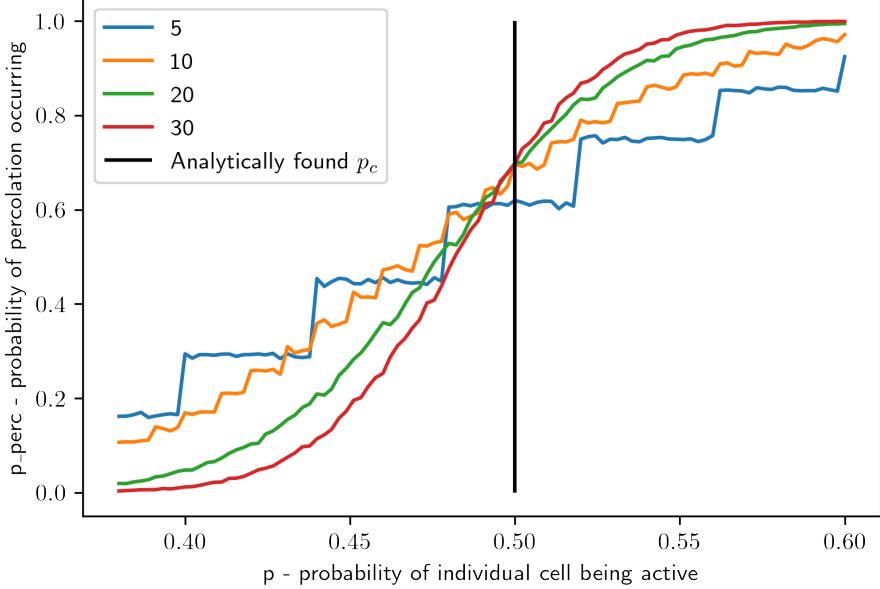


Figure 22: probability of percolation over the probability of individual cell activation, with analytically found critical probability ($p_c = 0.5$) highlighted.

Summary

Unfortunately, I did not have time to perform a renormalization analysis on the different 3D grids, although we can still take a holistic view over the analytically confirmed critical probabilities, and the empirically approximated ones.

In Figure 23 we can observe the decreasing fraction of cells that need to be activated p , for the $N = 30$ grids to have a 50% chance of percolation. We find that the RDG-6 grid is the one that percolates at the lowest value of cell activation probability p , and that the square grid is the one that needs the largest value.

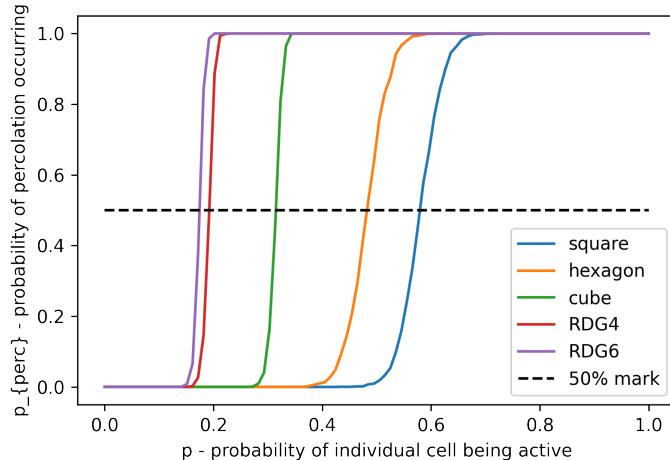


Figure 23: Percolation probability over cell activation probability for each type of grid, at $N = 30$.

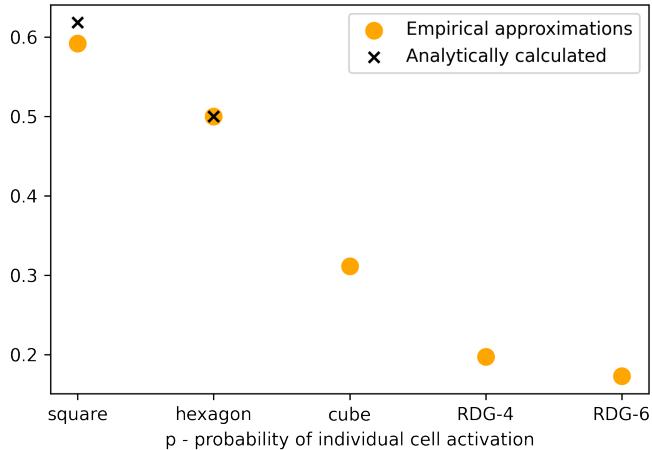


Figure 24: p_c , critical cell activation probability, for each type of grid. Compares empirical approximation with analytical results

References

- [1] Cyp, “Rhombic dodecahedron,” 03 2006. (under CC BY-NC-SA 3.0, has not been altered).
- [2] A. Kepert, “Rhombic dodecahedral honeycomb,” 07 2015. (under CC BY-NC-SA 3.0, has not been altered).
- [3] H. Sayama, “12.4: Renormalization group analysis to predict percolation thresholds,” 04 2018. (under CC BY-NC-SA 3.0, has been altered).
- [4] Z. V. Djordjevic, H. E. Stanley, and A. Margolina, “Site percolation threshold for honeycomb and square lattices,” *Journal of Physics A Mathematical and General*, vol. 15, pp. L405–L412, 08 1982.
- [5] J. L. Jacobsen, “High-precision percolation thresholds and potts-model critical manifolds from graph polynomials,” *Journal of Physics A*, vol. 47, pp. 135001–135001, 04 2014.
- [6] P. N. Suding and R. M. Ziff, “Site percolation thresholds for archimedean lattices,” *Physical Review E*, vol. 60, pp. 275–283, 07 1999.

2 Numerical solver for Laplace's Equation

We consider Laplace's equation

$$u_{xx} + u_{yy} = 0$$

on the square region $0 \leq x \leq \pi, 0 \leq y \leq \pi$, with the following boundary conditions:

$$u(0, y) = 0, u(\pi, y) = 0, u(x, 0) = 0, u(x, \pi) = \sin(3x).$$

Let's consider an $N \times N$ discretization grid, with indices $(1, 1)$ indicating the bottom-left point, and (N, N) the top-right point.

Remember that u satisfies Laplace's Equation:

$$u_{xx} + u_{yy} = 0$$

After discretization to a $N \times N$ grid, we can express the temperatures at the boundaries as:

$$u_{1,j} = 0 \quad \text{for } \{j \in N | 1 \leq j \leq N\}$$

$$u_{N,j} = 0 \quad \text{for } \{j \in N | 1 \leq j \leq N\}$$

$$u_{i,1} = 0 \quad \text{for } \{i \in N | 1 \leq i \leq N\}$$

$$u_{i,N} = \sin(3 \frac{i\pi}{N}) \quad \text{for } \{i \in N | 1 \leq i \leq N\}$$

Where the equation for $u_{i,N}$ is the only non-zero. We multiply with $\frac{\pi}{N}$ inside the sine function, since the discrete grid has the range $[1, N]$, but the boundary condition describes a range for the input that is $[0, \pi]$.

I can use a 5-point stencil to obtain an expression for u_{xx} and u_{yy} at all $(N - 2) \times (N - 2)$ internal points (that is, all points that are not on the boundaries). My expression for the derivatives at the (i, j) point will hence depend on the values $u_{ij}, u_{i+1,j}, u_{i-1,j}, u_{i,j+1}, u_{i,j-1}$.

To get approximations of the second derivative at a certain point we can make use of our discrete difference approximations. We do this by approximating our function with a Taylor series, cutting off the term after the one we want to find, and solving for our derivative. For example, to get an approximation of the second derivative at a location x_0 for a function $f(x)$: $f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \dots$

To keep track of the magnitude of the error, relative to the step size h , I will keep all terms up to the 4th derivative, and let the 4th derivative have an input ξ_1 where $\xi_1 \in (x, x + h)$.

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \frac{f'''(x_0)}{3!}(x - x_0)^3 + \frac{f^{(4)}(\xi_1)}{4!}(x - x_0)^4$$

Now if we have discrete points x_i and x_{i+1} , where the distance between points are a constant h , we can estimate the value at x_{i+1} and x_{i-1} using the Taylor series centered at x_i :

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i) + \frac{f''(x_i)}{2!}(x_{i+1} - x_i)^2 + \frac{f'''(x_i)}{3!}(x_{i+1} - x_i)^3 + \frac{f^{(4)}(\xi_1)}{4!}(x_{i+1} - x_i)^4$$

For x_{i+1} we get:

$$x_{i+1} = x_i + h$$

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_i + h - x_i) + \frac{f''(x_i)}{2!}(x_i + h - x_i)^2 + \frac{f'''(x_0)}{3!}(x_i + h - x_i)^3 + \frac{f^{(4)}(\xi_1)}{4!}(x_i + h - x_i)^4$$

$$f(x_{i+1}) = f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f'''(x_i)}{3!}h^3 + \frac{f^{(4)}(\xi_1)}{4!}h^4$$

For x_{i-1} , and using ξ_2 , where $\xi_2 \in (x_i - h, x_i)$, for the 4th derivative we get:

$$x_{i-1} = x_i - h$$

$$f(x_{i-1}) = f(x_i) + f'(x_i)(x_i - h - x_i) + \frac{f''(x_i)}{2!}(x_i - h - x_i)^2 + \frac{f'''(x_0)}{3!}(x_i - h - x_i)^3 + \frac{f^{(4)}(\xi_2)}{4!}(x_i - h - x_i)^4$$

$$f(x_{i-1}) = f(x_i) + f'(x_i)(-h) + \frac{f''(x_i)}{2!}(-h)^2 + \frac{f'''(x_i)}{3!}(-h)^3 + \frac{f^{(4)}(\xi_1)}{4!}(-h)^4$$

We can get an expression for the second derivative by summing the LHSs and RHSs for the Taylor's series of $f(x_{i+1})$ and $f(x_{i-1})$. It will now become apparent that the 3rd derivative cancels in this process, and that the 4th derivative is the first non-zero term that is cut off, and hence the largest error term.

$$\begin{aligned}
f(x_{i+1}) &= f(x_i) + f'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f'''(x_i)}{3!}h^3 + \frac{f^{(4)}(\xi_1)}{4!}h^4 \\
&\quad + \\
f(x_{i-1}) &= f(x_i) + f'(x_i)(-h) + \frac{f''(x_i)}{2!}(-h)^2 + \frac{f'''(x_i)}{3!}(-h)^3 + \frac{f^{(4)}(\xi_1)}{4!}(-h)^4 \\
&= \\
f(x_{i+1}) + f(x_{i-1}) &= 2f(x_i) + f'(x_i)(h - h) + 2\frac{f''(x_i)}{2!}h^2 + \frac{f'''(x_i)}{3!}(h^3 - h^3) + 2\frac{f^{(4)}(\xi_1)}{4!}h^4 \\
f''(x_i)h^2 &= f(x_{i+1}) + f(x_{i-1}) - 2f(x_i) - 2\frac{f^{(4)}(\xi_1)}{4!}h^4 \\
f''(x_i) &= \frac{f(x_{i+1}) + f(x_{i-1}) - 2f(x_i)}{h^2} - \frac{f^{(4)}(\xi_1)}{12}h^2
\end{aligned}$$

Our approximation is then:

$$f''(x_i) \approx \frac{f(x_{i+1}) + f(x_{i-1}) - 2f(x_i)}{h^2}$$

$$\text{We can now see that our error term: } \frac{f(x_{i+1}) + f(x_{i-1}) - 2f(x_i)}{h^2} - f''(x_i) = \frac{f^{(4)}(\xi_1)}{12}h^2$$

Scales with $O(h^2)$

$$f''(x_i) \approx \frac{f(x_{i+1}) + f(x_{i-1}) - 2f(x_i)}{h^2}$$

This is the central difference approximation for the second derivative for a function with a single independent variable. We can expand this to a discrete grid, where the step size for each independent variable is equal:

$$\Delta x = x_{i+1} - x_i = \Delta y = y_{i+1} - y_i = h$$

Where $u(x, y)$ is the function on the discrete grid, we can approximate the second derivative with respect to x by:

$$u_{xx} \Big|_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + O(h^2)$$

And the approximation for the second derivative with respect to y is derived with a change in y instead:

$$u_{yy} \Big|_{i,j} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2} + O(h^2)$$

We now have approximations for the partial derivatives u_{xx} and u_{yy} that are only in terms of the neighboring points. We can create a 5-point stencil to describe the relationship between the partial derivatives approximations and a linear combination of the neighborhood.

Where \circ is the Hadamard product (element-wise product), and $U_{3 \times 3}(i, j)$ is the 3×3 matrix of function evaluations centered around $u(i, j)$:

$$U_{3 \times 3}(i, j) = \begin{bmatrix} u(i-1, j+1) & u(i, j+1) & u(i+1, j+1) \\ u(i-1, j) & u(i, j) & u(i+1, j) \\ u(i-1, j-1) & u(i, j-1) & u(i+1, j-1) \end{bmatrix}$$

Then

$$u_{xx} \Big|_{i,j} = \frac{1}{h^2} \begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} \circ U_{3 \times 3}(i, j)$$

$$u_{yy} \Big|_{i,j} = \frac{1}{h^2} \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} \circ U_{3 \times 3}(i, j)$$

As analyzed above, we find that the central difference approximation for the second derivative has an error that scales with $O(h^2)$ for a step size of h in the grid. We have this error in both axes, although the compound effect of the error may only increase it with a scalar factor. If we were to calculate the 2-dimensional magnitude of the scaling of the error we would find:

$$|Error| = \sqrt{O(h^2)^2 + O(h^2)^2}$$

$$|Error| = \sqrt{2 \times O(h^4)}$$

$$|Error| = \sqrt{2} \times O(h^2)$$

$$|Error| = O(h^2)$$

Since scalar factors fall away in big-O notation, the asymptotic scaling of the error is still the same when we introduce a new dimension.

We can plug in our approximations for the second partial derivatives into Laplace's equation:

$$u_{xx} + u_{yy} = 0$$

$$u_{xx} \Big|_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2}$$

$$u_{yy} \Big|_{i,j} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2}$$

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2} = 0$$

$$\frac{1}{h^2}(u_{i-1,j} + u_{i+1,j} - 4u_{i,j} + u_{i,j-1} + u_{i,j+1}) = 0$$

Since $h \neq 0$

$$u_{i-1,j} + u_{i+1,j} - 4u_{i,j} + u_{i,j-1} + u_{i,j+1} = 0$$

$$u_{i,j} = \frac{u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}}{4}$$

If we instead express it in matrix form we get:

$$u_{xx} + u_{yy} \Big|_{i,j} = \frac{2}{h^2} \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \circ U_{3 \times 3}(i, j) = 0$$

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \circ U_{3 \times 3}(i, j) = 0$$

$$\begin{bmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & -1 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{bmatrix} \circ U_{3 \times 3}(i, j) = 0$$

$$u_{i,j} = \sum \begin{bmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{bmatrix} \circ U_{3 \times 3}(i, j)$$

Both versions express the same equation that needs to be satisfied for all inner $u_{i,j}$, where the qualitative interpretation of this is that every internal evaluation of $u_{i,j}$ needs to equal the average of its neighbors (in a Von Neumann neighborhood).

For the example where $N = 4$, with 2×2 internal points, we can express the equations that u_{22} , u_{23} , u_{32} , and u_{33} need to satisfy using the general formula:

$$u_{i-1,j} + u_{i+1,j} - 4u_{i,j} + u_{i,j-1} + u_{i,j+1} = 0$$

for u_{22} , u_{23} , u_{32} , and u_{33} :

$$u_{1,2} + u_{3,2} - 4u_{2,2} + u_{2,1} + u_{2,3} = 0$$

$$u_{1,3} + u_{3,3} - 4u_{2,3} + u_{2,2} + u_{2,4} = 0$$

$$u_{2,2} + u_{4,2} - 4u_{3,2} + u_{3,1} + u_{3,3} = 0$$

$$u_{2,3} + u_{4,2} - 4u_{3,3} + u_{3,2} + u_{3,4} = 0$$

We know the boundary conditions:

$$u_{1,j} = 0 \quad \text{for } \{j \in N | 1 \leq j \leq N\}$$

$$u_{N,j} = 0 \quad \text{for } \{j \in N | 1 \leq j \leq N\}$$

$$u_{i,1} = 0 \quad \text{for } \{i \in N | 1 \leq i \leq N\}$$

$$u_{i,N} = \sin(3 \frac{i\pi}{N}) \quad \text{for } \{i \in N | 1 \leq i \leq N\}$$

So after substituting we have:

$$0 + u_{3,2} - 4u_{2,2} + 0 + u_{2,3} = 0$$

$$0 + u_{3,3} - 4u_{2,3} + u_{2,2} + \sin(3 \frac{2\pi}{4}) = 0$$

$$u_{2,2} + 0 - 4u_{3,2} + 0 + u_{3,3} = 0$$

$$u_{2,3} + 0 - 4u_{3,3} + u_{3,2} + \sin(3\frac{2\pi}{4}) = 0$$

$$u_{3,2} - 4u_{2,2} + u_{2,3} = 0$$

$$u_{3,3} - 4u_{2,3} + u_{2,2} = -\sin(\frac{3}{2}\pi) = 1$$

$$u_{2,2} - 4u_{3,2} + u_{3,3} = 0$$

$$u_{2,3} - 4u_{3,3} + u_{3,2} = -\sin(\frac{9}{4}\pi) = -\frac{1}{\sqrt{2}}$$

We can set up the vector \vec{v} for the RHS's:

$$\vec{v} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$$

The unknowns \vec{u} :

$$\vec{u} = \begin{bmatrix} u_{22} \\ u_{23} \\ u_{32} \\ u_{33} \end{bmatrix}$$

And the coefficient matrix A :

$$A = \begin{bmatrix} -4 & 1 & 1 & 0 \\ 1 & -4 & 0 & 1 \\ 1 & 0 & -4 & 1 \\ 0 & 1 & 1 & -4 \end{bmatrix}$$

Such that the matrix representation for the system of equations is:

$$\begin{bmatrix} -4 & 1 & 1 & 0 \\ 1 & -4 & 0 & 1 \\ 1 & 0 & -4 & 1 \\ 0 & 1 & 1 & -4 \end{bmatrix} \begin{bmatrix} u_{22} \\ u_{23} \\ u_{32} \\ u_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$$

$$A\vec{u} = \vec{v}$$

We can solve the system of equations $A\vec{u} = \vec{v}$ by left-multiplying both sides with the inverse of our coefficient matrix A^{-1} .

We find that A^{-1} equals:

$$A^{-1} = -\frac{1}{12} \begin{bmatrix} 3.5 & 1 & 1 & 0.5 \\ 1 & 3.5 & 0.5 & 1 \\ 1 & 0.5 & 3.5 & 1 \\ 0.5 & 1 & 1 & 3.5 \end{bmatrix}$$

Which allows us to find \vec{u} :

$$A^{-1}A\vec{u} = A^{-1}\vec{v}$$

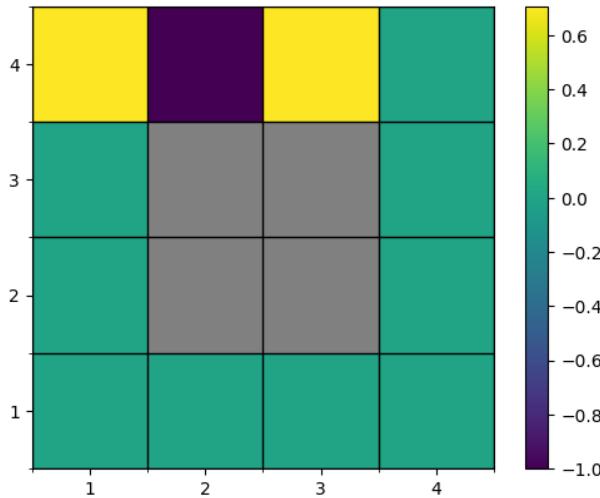
$$I\vec{u} = A^{-1}\vec{v}$$

$$\vec{u} = A^{-1}\vec{v}$$

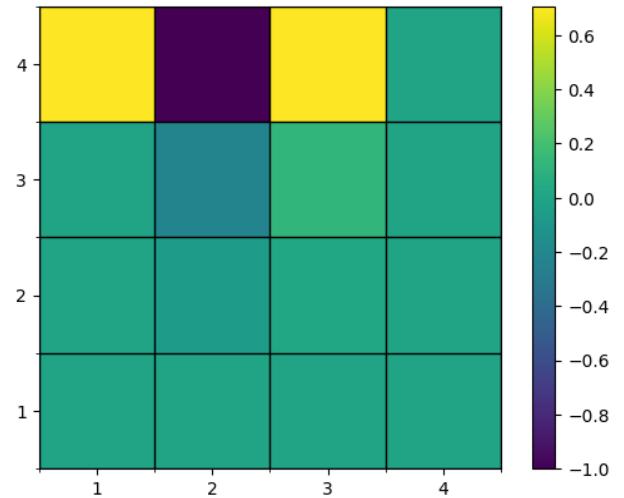
$$A^{-1}\vec{v} = -\frac{1}{12} \begin{bmatrix} 3.5 & 1 & 1 & 0.5 \\ 1 & 3.5 & 0.5 & 1 \\ 1 & 0.5 & 3.5 & 1 \\ 0.5 & 1 & 1 & 3.5 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ -\frac{1}{\sqrt{2}} \end{bmatrix} = -\frac{1}{12} \begin{bmatrix} 1 - \frac{1}{2\sqrt{2}} \\ 3.5 - \frac{1}{\sqrt{2}} \\ 0.5 - \frac{1}{\sqrt{2}} \\ 1 - \frac{3.5}{\sqrt{2}} \end{bmatrix} \approx \begin{bmatrix} -0.05387 \\ -0.23274 \\ 0.01726 \\ 0.12291 \end{bmatrix}$$

We now know the values of the inner points \vec{u} :

$$\vec{u} = \begin{bmatrix} u_{22} \\ u_{23} \\ u_{32} \\ u_{33} \end{bmatrix} = -\frac{1}{12} \begin{bmatrix} 1 - \frac{1}{2\sqrt{2}} \\ 3.5 - \frac{1}{\sqrt{2}} \\ 0.5 - \frac{1}{\sqrt{2}} \\ 1 - \frac{3.5}{\sqrt{2}} \end{bmatrix}$$



(a) Boundary conditions for 4x4 matrix

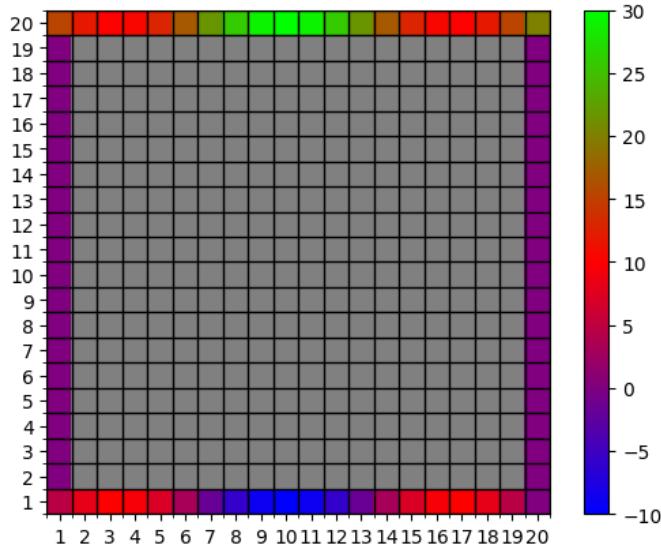


(b) 4x4 matrix with analytical approximations in inner cells

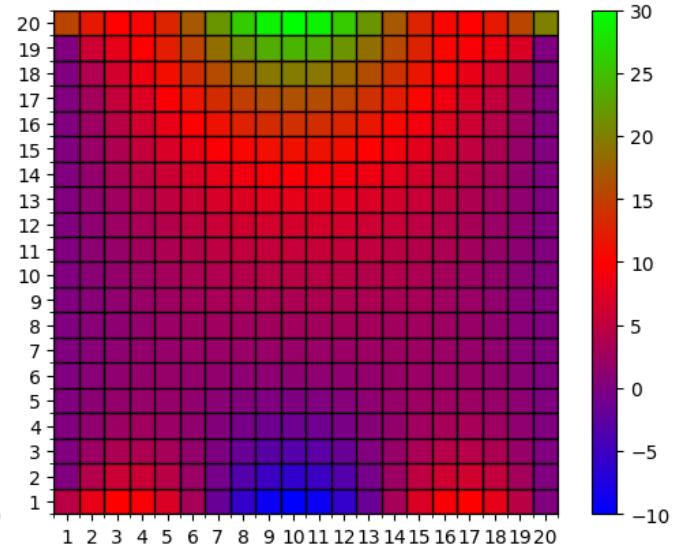
Figure 25: To the left (a) we see a visual representation of the boundary conditions, where all edges are at 0, except the top that follows $\sin(3\frac{i}{N}\pi)$. To the right (b) we see the complete grid after filling in the analytical approximations we found above (\vec{u}).

I made python code that can take in any arbitrary width for the target grid (N^2), and any arbitrary functions for the boundary conditions. The code numerically calculates the approximations for each inner cell, such that it satisfies Laplace's equation. It does this through the same method as above, by creating the coefficient matrix A that represents the 5-point stencil that is consistent with Laplace's equation, and the answer vector \vec{v} . Finally solving for \vec{u} by computing $A^{-1}\vec{v}$.

Below are 3 examples of what the code outputted.

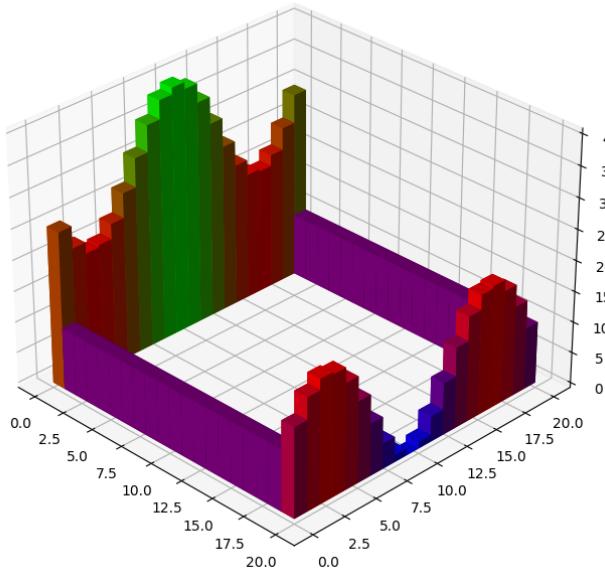


(a) 2D representation of boundary conditions for 20x20 matrix

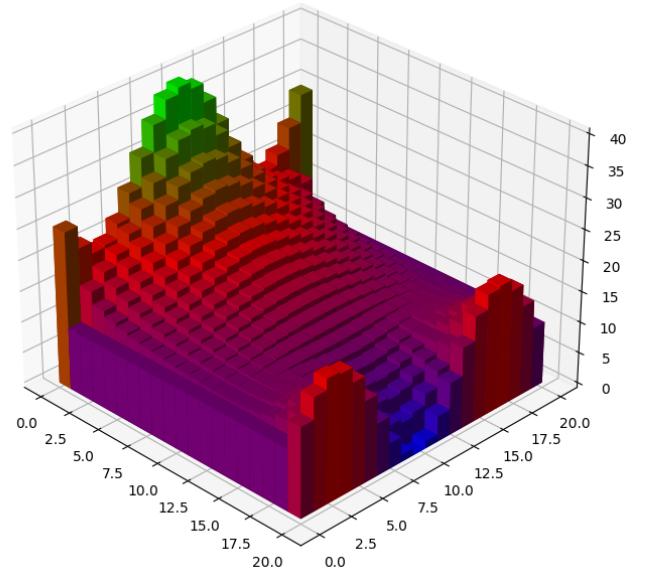


(b) Same matrix after filling in the numerical approximations

Figure 26: Example 1. A 20x20 grid with boundary conditions: $u_{x,20} = 20 - 10\sin(3\pi x)$, $u_{x,1} = 10\sin(3\pi x)$, $u_{1,y} = 0$, $u_{20,y} = 0$.

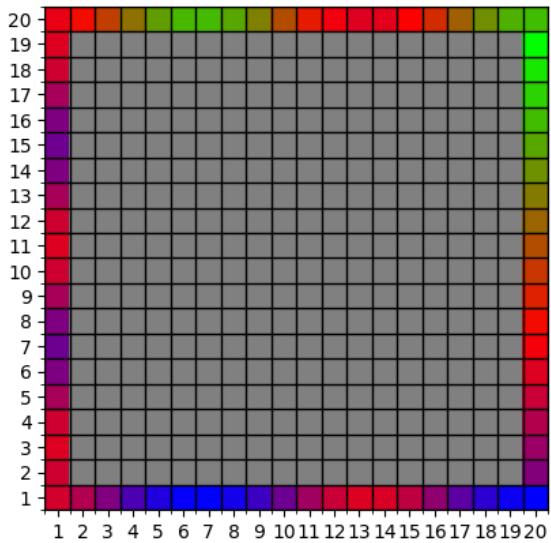


(a) 3D representation of boundary conditions for 20×20 matrix

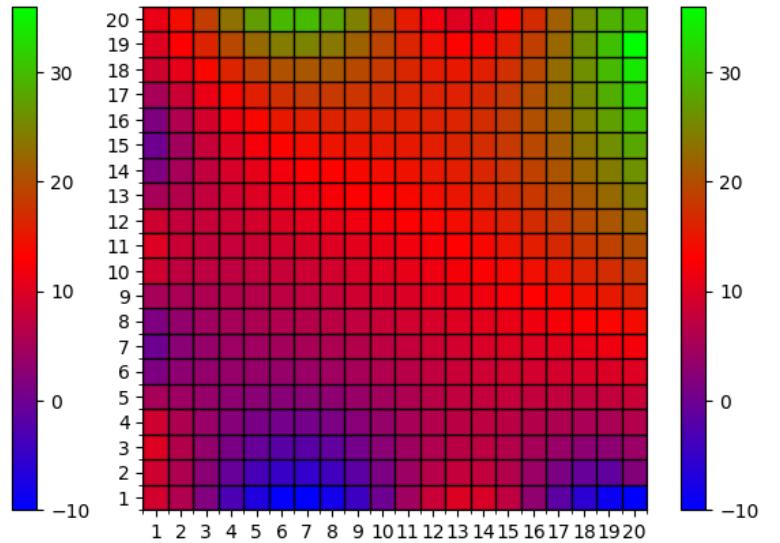


(b) Same matrix after filling in the numerical approximations

Figure 27: Example 1. The same system as in Figure 26a, but visualized in 3D by letting height represent function evaluation.

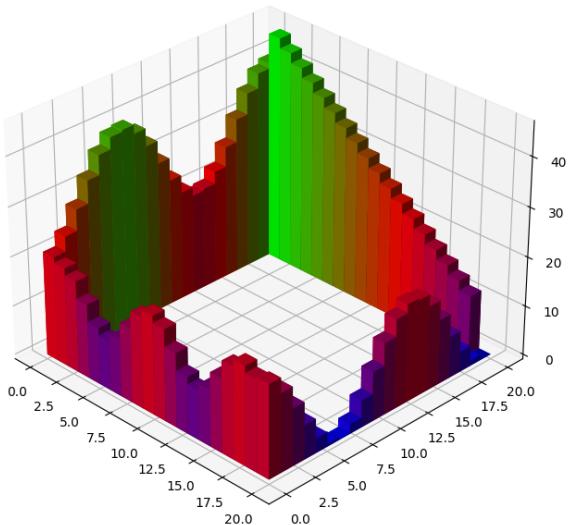


(a) 2D representation of boundary conditions for 20×20 matrix

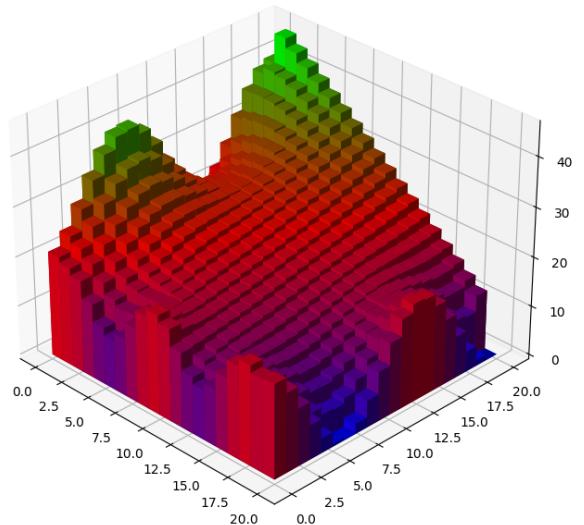


(b) Same matrix after filling in the numerical approximations

Figure 28: Example 2. A 20×20 grid with boundary conditions: $u_{x,20} = 20 - 10\cos(3\pi x)$, $u_{x,1} = 10\cos(3\pi x)$, $u_{1,y} = 5 + 5\sin(5\pi y)$, $u_{20,y} = 40 - 40y$.

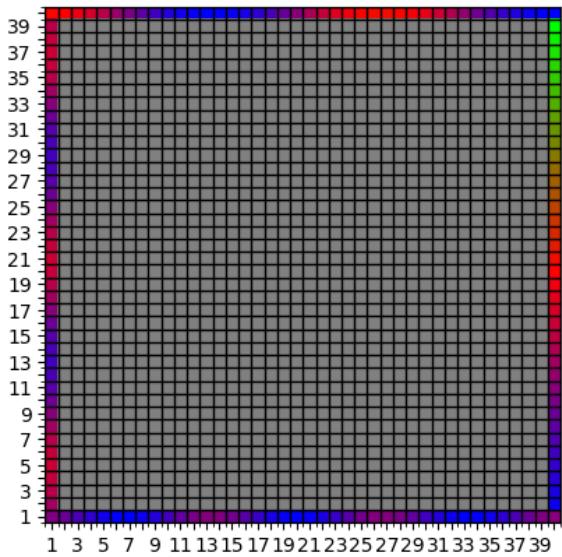


(a) 3D representation of boundary conditions for 20×20 matrix

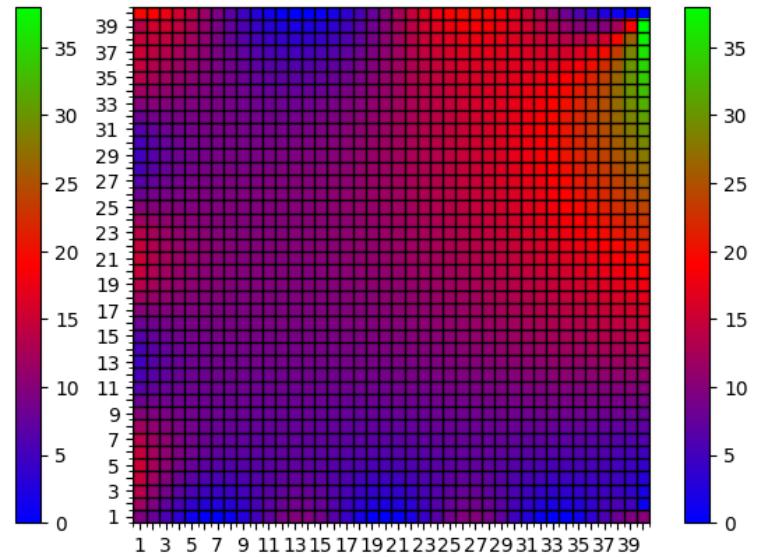


(b) Same matrix after filling in the numerical approximations

Figure 29: Example 2. The same system as in Figure 28a, but visualized in 3D by letting height represent function evaluation.



(a) 2D representation of boundary conditions for 40×40 matrix



(b) Same matrix after filling in the numerical approximations

Figure 30: Example 3. A 40×40 grid with boundary conditions: $u_{x,20} = 10 + 10\cos(3\pi x)$, $u_{x,1} = 5 + 5\cos(6\pi x)$, $u_{1,y} = 10 + 5\sin(5\pi y)$, $u_{20,y} = 40 - 40y$.

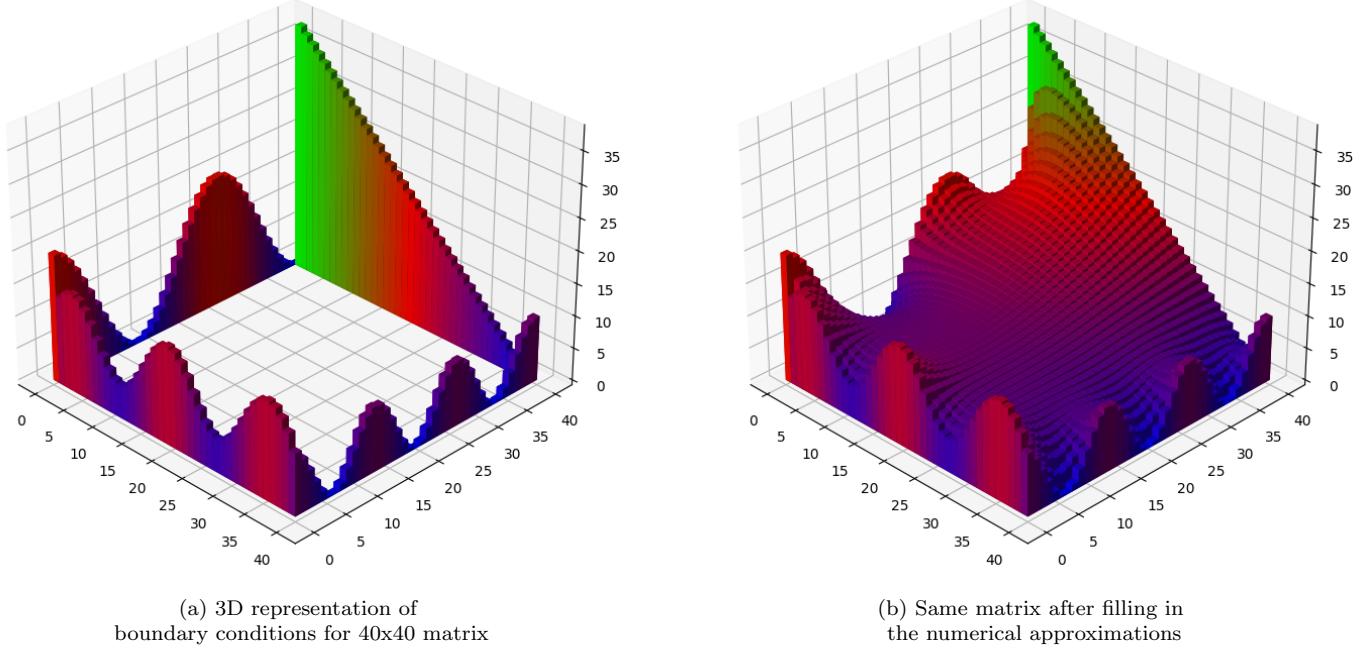


Figure 31: Example 3. The same system as in Figure 30a, but visualized in 3D by letting height represent function evaluation.

If we instead have the Poisson Equation, which is a forced (non-homogeneous) version of Laplace's Equation:
 $u_{xx} + u_{yy} = p(x, y)$

We would still use the central difference approximations for the second derivatives:

$$u_{xx} \Big|_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2}$$

$$u_{yy} \Big|_{i,j} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2}$$

But instead of plugging it into Laplace's Equation, we would plug it into the Poisson Equation:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2} \Big|_{i,j} = p(x, y)$$

$$u_{i+1,j} + u_{i-1,j} - 4u_{i,j} + u_{i,j+1} + u_{i,j-1} \Big|_{i,j} = h^2 p(x, y)$$

Now we instead have the non-homogeneous system of equations:

$$u_{1,2} + u_{3,2} - 4u_{2,2} + u_{2,1} + u_{2,3} = h^2 p(x, y)$$

$$u_{1,3} + u_{3,3} - 4u_{2,3} + u_{2,2} + u_{2,4} = h^2 p(x, y)$$

$$u_{2,2} + u_{4,2} - 4u_{3,2} + u_{3,1} + u_{3,3} = h^2 p(x, y)$$

$$u_{2,3} + u_{4,2} - 4u_{3,3} + u_{3,2} + u_{3,4} = h^2 p(x, y)$$

Where we now can plug in our known values:

$$u_{1,j} = 0 \quad \text{for } \{j \in N | 1 \leq j \leq N\}$$

$$u_{N,j} = 0 \quad \text{for } \{j \in N | 1 \leq j \leq N\}$$

$$u_{i,1} = 0 \quad \text{for } \{i \in N | 1 \leq i \leq N\}$$

$$u_{i,N} = \sin(3 \frac{i\pi}{N}) \quad \text{for } \{i \in N | 1 \leq i \leq N\}$$

$h = 1$ (Each square is considered to be one unit away from each other in our grid)

$$p(x, y) = x^2$$

$$0 + u_{3,2} - 4u_{2,2} + 0 + u_{2,3} = 1^2 x^2$$

$$0 + u_{3,3} - 4u_{2,3} + u_{2,2} + \sin(3 \frac{2}{4}\pi) = 1^2 x^2$$

$$u_{2,2} + 0 - 4u_{3,2} + 0 + u_{3,3} = 1^2 x^2$$

$$u_{2,3} + 0 - 4u_{3,3} + u_{3,2} + \sin(3\frac{3}{4}\pi) = 1^2 x^2$$

$$u_{3,2} - 4u_{2,2} + u_{2,3} = x^2$$

$$u_{3,3} - 4u_{2,3} + u_{2,2} = x^2 + 1$$

$$u_{2,2} - 4u_{3,2} + u_{3,3} = x^2$$

$$u_{2,3} - 4u_{3,3} + u_{3,2} = x^2 - \frac{1}{\sqrt{2}}$$

Here we remember that the first equation is the evaluation at coordinate $(2, 2)$, hence $x = 2$, and the 3rd row is at coordinate $(3, 2)$, hence $x = 3$.

$$u_{3,2} - 4u_{2,2} + u_{2,3} = 2^2 = 4$$

$$u_{3,3} - 4u_{2,3} + u_{2,2} = 2^2 + 1 = 5$$

$$u_{2,2} - 4u_{3,2} + u_{3,3} = 3^2 = 9$$

$$u_{2,3} - 4u_{3,3} + u_{3,2} = 3^2 - \frac{1}{\sqrt{2}} = 9 - \frac{1}{\sqrt{2}}$$

We can now solve this system of equations just like above, simply with different values in our \vec{v} :

$$\vec{v} = \begin{bmatrix} 4 \\ 5 \\ 9 \\ 9 - \frac{1}{\sqrt{2}} \end{bmatrix}, \vec{u} = \begin{bmatrix} u_{22} \\ u_{23} \\ u_{32} \\ u_{33} \end{bmatrix}, A = \begin{bmatrix} -4 & 1 & 1 & 0 \\ 1 & -4 & 0 & 1 \\ 1 & 0 & -4 & 1 \\ 0 & 1 & 1 & -4 \end{bmatrix}$$

$$A\vec{u} = \vec{v}$$

$$\vec{u} = A^{-1}\vec{v}$$

$$\vec{u} = -\frac{1}{12} \begin{bmatrix} 3.5 & 1 & 1 & 0.5 \\ 1 & 3.5 & 0.5 & 1 \\ 1 & 0.5 & 3.5 & 1 \\ 0.5 & 1 & 1 & 3.5 \end{bmatrix} \begin{bmatrix} 4 \\ 5 \\ 9 \\ 9 - \frac{1}{\sqrt{2}} \end{bmatrix}$$

$$\vec{u} \approx \begin{bmatrix} -2.30387 \\ -2.10774 \\ -3.10774 \\ -1.12709 \end{bmatrix}$$

3 The Trace-Determinant Plane

When we have a system of differential equations of the form

$$x' = ax + by$$

$$y' = cx + dy$$

We can express it as a matrix multiplication by letting \vec{x} be a vector of our functions x and y , and \vec{x}' be a vector of their first derivatives (it can be assumed it's in respect to time).

$$\vec{x} = \begin{bmatrix} x \\ y \end{bmatrix}, \vec{x}' = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

This allows us to write a coefficient matrix A such that:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\vec{x}' = A\vec{x}$$

Notice that the system is completely defined by A , our coefficients.

To analyze the qualitative behavior of our system we would plot a direction field, where you let one axis be the value of function x , the other axis be the value of y , and the vector at that point be the derivative in respect to time.

Where your system is in the form:

$$\frac{dx}{dt} = f(x, y)$$

$$\frac{dy}{dt} = g(x, y)$$

The vector at a point (x_0, y_0) points in the direction $(f(x_0, y_0), g(x_0, y_0)) = (\frac{dx}{dt}, \frac{dy}{dt})$

Here are some examples of coefficient matrices and their behavior displayed in quiver plots.

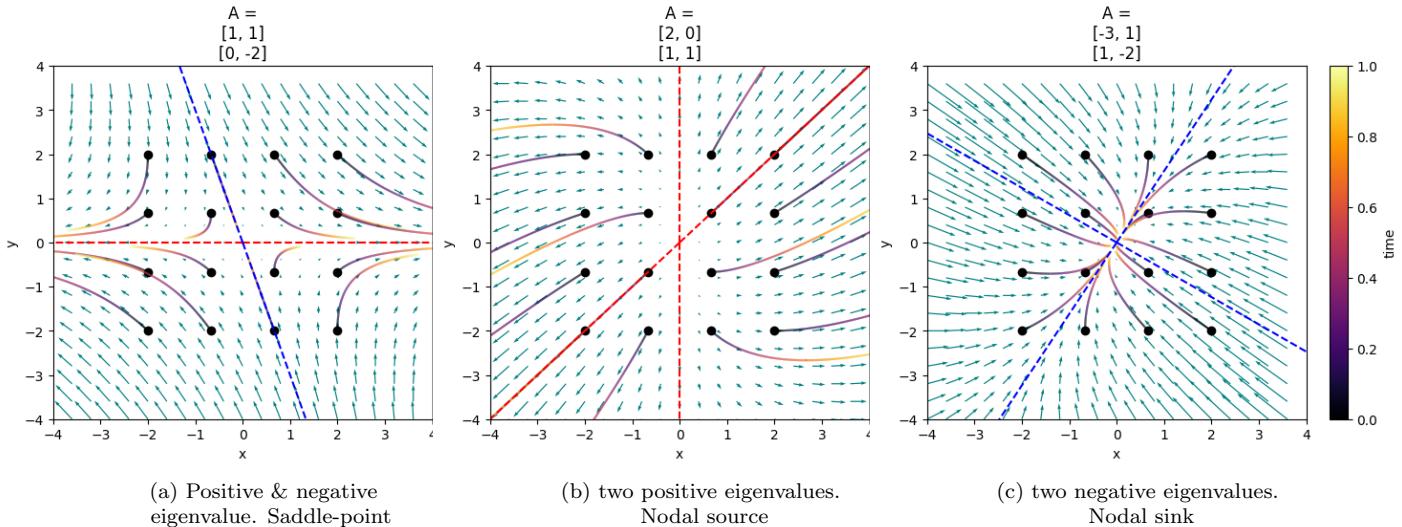


Figure 32: Three examples of coefficient matrices, and their behavior in direction-plots. Red dashed lines correspond to eigenvectors with positive eigenvalues, Blue dashed lines correspond to eigenvectors with negative eigenvalues. The curves correspond to solutions over time given different starting points (initial conditions).

Above we see the behavior of different systems of linear ODEs. The dashed lines are both the eigenvectors and stable solution lines, meaning that any solution that starts on one of the lines stays there, for example, the diagonal dots in Figure 32b where 4 dots align with the diagonal eigenvector and stay on it over time. For all three sub-figures, we see that the solution space gets divided into 4 regions since solutions can't cross the stable-solution lines (eigenvectors). We can also see that there is a qualitative difference in whether the

eigenvalues for the eigenvectors are positive or negative. When looking at the plots, notice that solutions are scaled in the direction of their eigenvectors with the magnitude of the corresponding eigenvalues. If we were to express the coefficient matrix A in its eigenbasis, we would get a diagonal matrix with the eigenvalues in the diagonal, which represents the axis-wise scaling where the axes are the eigenvectors.

In Figure 32a, we have one positive and one negative eigenvalue, solutions go towards 0 in the perspective of the blue (negative) eigenvector, and away from 0 in the perspective of the red (positive) eigenvector. The origin $(0, 0)$ is a stationary stable state, where every solution starting on the blue eigenvector will over time approach it, any other initial condition will not reach the stationary state.

In Figure 32b, we have two positive eigenvalues, all solutions go away from the origin everywhere. The origin $(0, 0)$ is a stationary stable state, where only the solution starting on the origin will reach it, any other initial condition will move away from the stationary state over time.

In Figure 32c, we have two negative eigenvalues, all solutions go towards the origin everywhere. The origin $(0, 0)$ is a stationary stable state, where all solutions move towards it.

Some coefficient matrices A will lead to a double root and a single real-valued eigenvalue. For example:

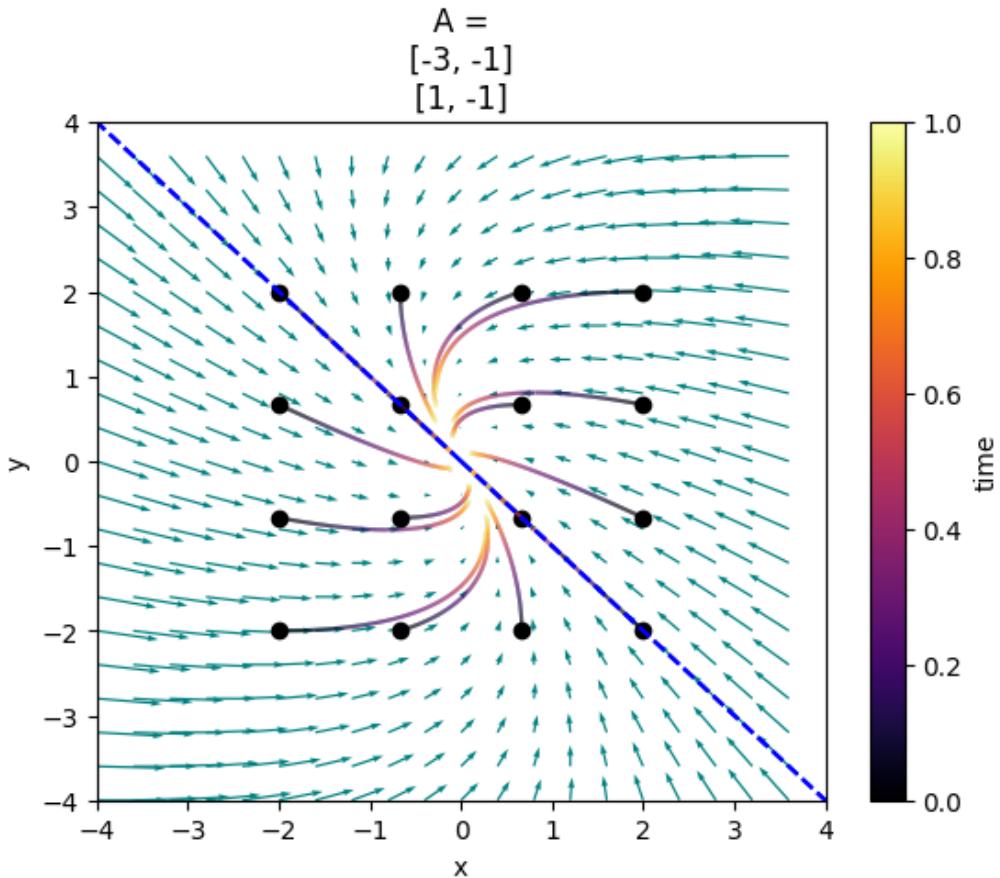


Figure 33: A single real-valued eigenvalue, there is a single stable solution-line.

Some coefficient matrices A will lead to complex-valued eigenvectors. For example:

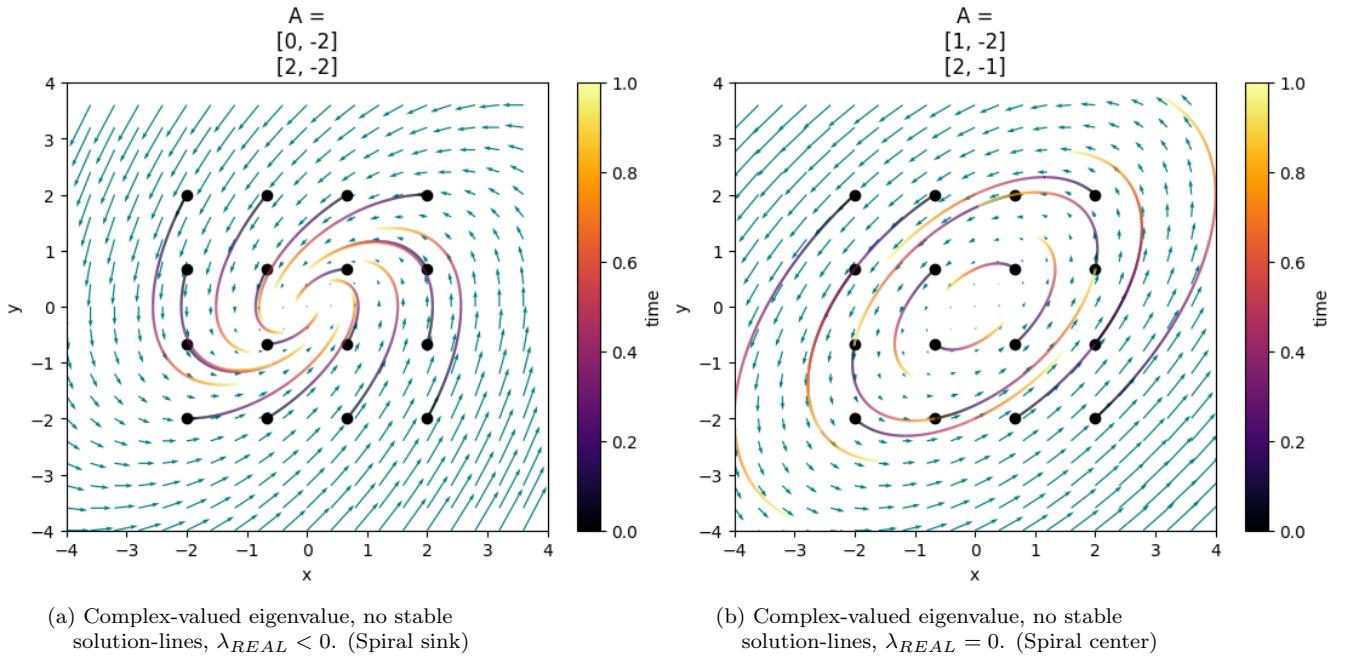


Figure 34: Two examples of coefficient matrices where we get complex eigenvalues. Sub-figure (a) shows how the behavior looks like when we have an eigenvalue with a negative real component in the eigenvalue, all solutions tend to the origin, which is a stable stationary state. Sub-figure (b) shows how the behavior looks like when we have an eigenvalue with a real component exactly equal to 0, all solutions stay on their elliptical orbit.

In the figure above, solutions will still approach the origin, which still is a stationary state. But we don't see any stable solution lines. We can analyze exactly when a given coefficient matrix A will have a single point that is stationary over time (Figure 32b), a line which will lead to a stationary state (Figure 32a), or where the entire plane leads to a stationary state (Figure 32c & Figure 33). We do this by tracking how the trace and determinant of A impacts the eigenvalues of A .

To find the eigenvalues of A , we find when the determinant of $A - \lambda I$ equals 0, where λ is the eigenvalue and I is the identity matrix.

$$\det(A - \lambda I) = \det \begin{pmatrix} a - \lambda & b \\ c & d - \lambda \end{pmatrix} = (a - \lambda)(d - \lambda) - bc$$

$$(a - \lambda)(d - \lambda) - bc = \lambda^2 - (a + d)\lambda + (ad - bc)$$

Where the trace $tr(A)$ is:

$$\tau = a + d$$

And the determinant $\det(A)$ is:

$$\Delta = ad - bc$$

This lets us rewrite the characteristic polynomial as:

$$\det(A - \lambda I) = \lambda^2 - \tau\lambda + \Delta$$

If we solve for the eigenvalues λ :

$$\lambda = \frac{\tau \pm \sqrt{\tau^2 - 4\Delta}}{2}$$

Now it is revealed that the discriminant $\tau^2 - 4\Delta$ dictates whether our eigenvalues will be real or complex. Specifically:

If $\tau^2 - 4\Delta > 0$, we have two distinct and real eigenvalues.

If $\tau^2 - 4\Delta = 0$, we have a double root, two repeated real eigenvalues ($\frac{\tau}{2}$ is real).

If $\tau^2 - 4\Delta < 0$, we have two distinct and complex eigenvalues.

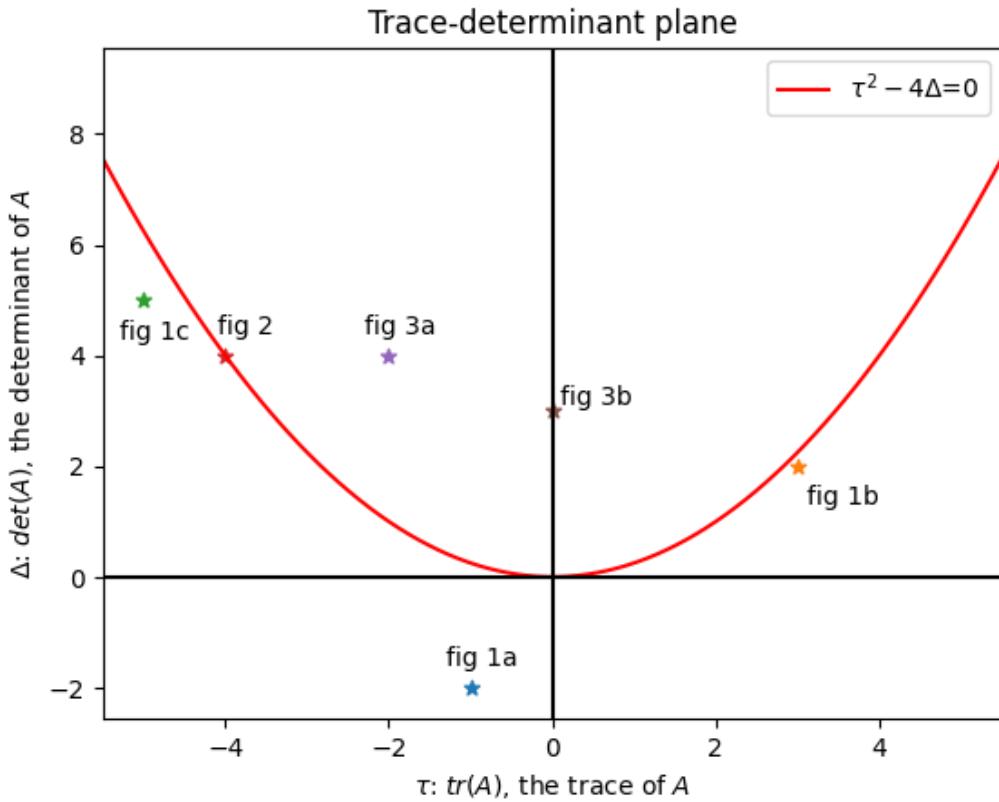


Figure 35: Trace-determinant plot, where the discriminant equals 0 is highlighted in red. The stars are the specific positions for the different examples above.

If we plot the trace against the determinant of a given matrix and plot the boundary where $\tau^2 - 4\Delta = 0$ we will find that the qualitative properties of the system behavior depend on where the coefficient matrix is in the trace-determinant plane.

If we're below the $\tau^2 - 4\Delta = 0$ line, we have two real eigenvalues (Figure 32a, 32b, 32c).

If we're exactly on the line, we have only one real eigenvalue, so only one stable line (Figure 33).

If we're above the line ($\tau^2 - 4\Delta < 0$), we have complex-valued eigenvalues (Figure 34a, 34b).

It also matters whether the determinant Δ is larger or less than 0. Looking at the quadratic equation for finding the eigenvalues, we can see that if we have a positive determinant $\Delta > 0$:

It's true that:

$$|\tau| > \sqrt{\tau^2 - 4\Delta}$$

since

$$\tau^2 > \tau^2 - 4\Delta \text{ when } \Delta > 0$$

If $\tau > 0$, $|\tau| = \tau$ and $\tau \pm \sqrt{\tau^2 - 4\Delta} > 0$ for both + and - when calculating the eigenvalues.

$$\lambda = \frac{\tau \pm \sqrt{\tau^2 - 4\Delta}}{2}$$

$$\lambda_1 > 0 \text{ and } \lambda_2 > 0$$

But if $\tau < 0$, $\tau = -|\tau|$ and $-|\tau| \pm \sqrt{\tau^2 - 4\Delta} < 0$ for both + and - when calculating the eigenvalues.

$$\lambda = \frac{-\tau \pm \sqrt{\tau^2 - 4\Delta}}{2}$$

$$\lambda_1 < 0 \text{ and } \lambda_2 < 0$$

If instead $\Delta < 0$

$$\tau < \sqrt{\tau^2 - 4\Delta}$$

since

$$\tau^2 < \tau^2 - 4\Delta \text{ when } \Delta < 0$$

This means that

$$\tau + \sqrt{\tau^2 - 4\Delta} > 0$$

$$\tau - \sqrt{\tau^2 - 4\Delta} < 0$$

$$\lambda_1 > 0 \text{ and } \lambda_2 < 0$$

We have found that if $\Delta < 0$, we have one positive and one negative eigenvalue. If $\Delta > 0$ and $\tau > 0$, we

have two positive eigenvalues, and if $\Delta > 0$ and $\tau < 0$, we have two negative eigenvalues. This aligns fully with Figure 35, where we can see that fig 1a is in $\Delta < 0$ and has one positive and one negative eigenvalue, and fig 1b with two positive eigenvalues is in $\Delta > 0$ and $\tau > 0$, and fig 1c with two negative eigenvalues is in $\Delta > 0$ and $\tau < 0$. We can visualize these different regions and their qualitative meaning.

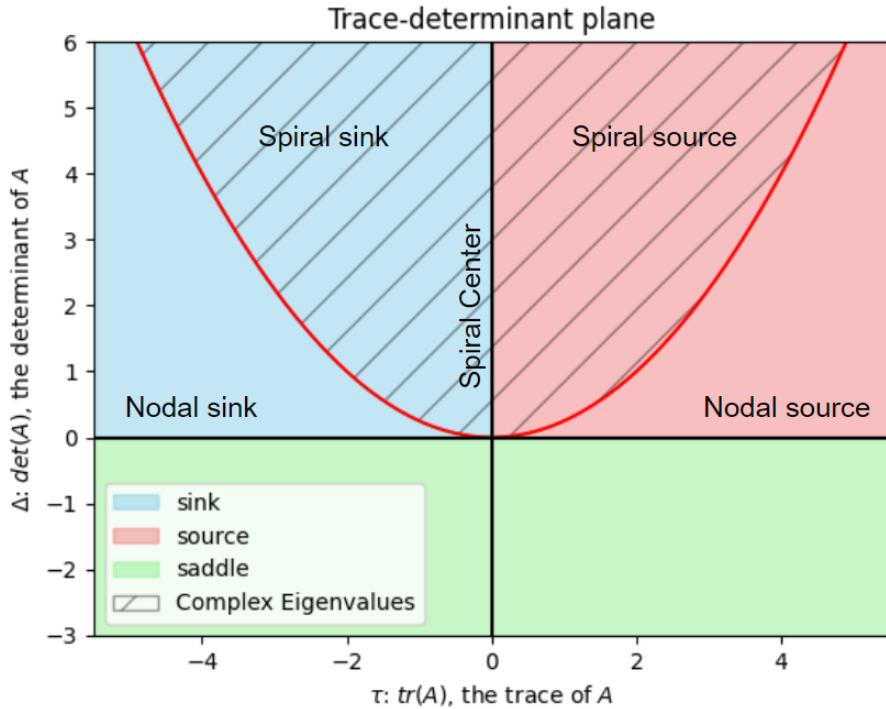


Figure 36: Trace-determinant plot, where the discriminant equals 0 is highlighted in red. Regions of qualitative difference are highlighted in different colors, and complex eigenvalues are in the striped region

Figure 36 neatly summarizes the different regions that exist on the trace-determinant plane, and their qualitative interpretation. Notice that every system that is exactly on the line red line corresponds with exactly one eigenvalue, and hence a single stable line. For example, Figure 33 has a single stable line and is a sink, which is why it would land on the line and in the blue-shaded region. Also, notice the "Spiral Center" line where $\Delta > 0$ and $\tau = 0$. This line corresponds to complex eigenvalues, which are neither a source nor a sink, but where solutions instead travel on stable ellipses (Figure 34b).