Scientific Computing Exercise Set 2

Thomas Norton (15295273) & Balthazar Dupuy d'Angeac (15852431) & Ruben Lanjouw (12415049)

***Repository**: https://github.com/RSlanjouw/Scientific-Computing*

INTRODUCTION

In order to accurately model a natural system, one can not work around simulating varying processes contrasting with and challenging one another. Physical reactions like *diffusion* usually unravel simultaneously to other forces, like the diffusion of a different element or species. Simulating conflicting physical reactions within the same system separately is not possible in most cases; they are often intertwined. Say one were to model chess; they could not compute all white moves and then black moves; both processes depend on and update the status of the board -of the system. How we might model this relates to a core idea behind integrating conflicting discretized processes: one updates the system for a short time step δt then the other process does: by turn. The smaller the step, the more fluid the interaction is. This is one way to re-conciliate two conflicting natural processes; where we juggle between two or more interactions/equations, indulging them one at the time. Yet we hope to -by way of understanding their relationship- produce an overarching, and often more complex, equation. One which, on it's own, might define the system and therefore, our model. In this paper, we experiment on how we may model such a system; defined by two distinct physical processes. We first examine a *Diffusion Limited Aggregation* (DLA) model; determined by diffusion of particles along side a growing insulated object. This object grows depending on the concentration of diffused particles in its surroundings. We first model this system indulging both diffusion and the cluster's growth separately, periodically. We then attempt to model it under a single governing process through the Monte Carlo approach. Finally, we explore the *Gray-Scott* model, determined by the interaction between the reaction and diffusion of two chemicals. Using different parameters, we examine conditions in which stable or chaotic structures emerge.

I. THEORY

A. Diffusion Limited Aggregation

Diffusion Limited Aggregation consists of the combination of two dynamics: diffusion and aggregation. Diffusion is the process by which particles or material properties, such as heat or gas, move through a medium. Aggregation is the process by which particles assign themselves to a cluster; this can sometimes lead to fractal growth patterns. In practice, we observe this in the growth of a bacterial colony, but also in the expansion of cities. The DLA depends mathematically on two equations:

1) *Diffusion Equation*, described by the time-independend Laplace-equation given as:

$$D\nabla^2 c = 0 \tag{1}$$

Implementing a discretized version of 1 requires starting conditions given by $C(x, y = 0; t) = 1$ and $C(x, y = 1; t) = 0$ and periodic boundaries. There are different iteration methods to implement this, whereby SOR is the most efficient[1].

2) *Aggregation*'s dynamic follows

$$v_n = -\frac{D}{\rho_0}\frac{\partial c}{\partial n} \tag{2}$$

where n is the normal direction of the surface, ρ_0 is the concentration needed for aggregating and D is the diffusion coefficient from 1. We implemented the probability of growth at a given site as:

$$p_0(i,j) = \frac{c_{i,j}^{\eta}}{\sum_{\text{growth candidates}} c_{i,j}^{\eta}}, \tag{3}$$

The η parameter is a scaling factor, and c represents the concentration at its position. This formulation ensures that at each iteration, a new site is incorporated into the cluster, as the denominator accounts for the cumulative concentration of all possible growth sites, making the chance on expansion 100%.

B. Monte-Carlo Simulation Approach

The Monte-Carlo approach to modelling serves to fill in a significant gap in our knowledge of the studied system.

When simulating an existing system, we often -if not always- have uncertainties about the system: we find parameters which behave in ways we fail to capture in equations/dynamics we can integrate. While we may not

know what physical rules govern such a parameter (say, μ), we sometimes know how μ is *expected* to behave. We can therefore attempt to approximate this parameter, or proceed without knowing how it behaves with precision, by testing our system for a significantly high number of uniformly random replacements for μ.

Our selection of replacements may be sharpened by potential knowledge we may have of the parameter's boundaries. As we run these numerous tests, we underline when the random parameters trigger an expected change in the system. Those values we save and disregard the others. This is the **Monte Carlo** approach and it works as a decent approximation of an unknown parameter or behaviour, thanks in part to the *Law of Large Numbers*, because the massive random sample gives us a big subset of random parameters which triggered the system in a desired manner, whatever we can derive from this successful sub-sample we can extrapolate of our parameter μ.

Take a standard example: deriving π. No one knows the exact value of π, yet we can approximate it with Monte Carlo; shooting 'darts' randomly on a square sheet of paper. We can determine the exact centre of the square, then count the darts that fell within distance r of the centre. Adding those up gives us an approximation of the portion of area A of a circle of radius r with contrast to the total amount of thrown darts; $\frac{darts\ within\ radius}{total\ darts} \approx A$ and $A = \pi r^2 \implies \pi = \frac{A}{r^2}$, giving us an approximation for π. The bigger the sample of darts, the more accurate the approximation is.

We do not know exactly how particles spread and stick to an object, depending on this object's shape. But we can assume that the particle sticks when it comes into contact with the object and that the particle moves around. We will examine what object is shaped by a massive sample of randomly walking particles.

C. Gray-Scott Model

The Gray-Scott model is a reaction diffusion system which describes the interaction of two chemicals U and V as they diffuse and react with each other. The reaction rate at each point can be determined by the local concentration of the chemicals. The reactions are:

$$\begin{aligned} U + 2V &\to 3V \\ V &\to P \end{aligned} \tag{4}$$

The chemical U is continuously added to the system, where it reacts with V to produce more of V. At the same time, V breaks down into P, a chemical that does not interact with either U or V.

Letting u and v represent the concentrations of U and V, the system can be described by the following equations:

$$\begin{aligned} \frac{\partial u}{\partial t} &= D_u \nabla^2 u \ - \ u\,v^2 \ + \ f\,(1-u), \\ \frac{\partial v}{\partial t} &= D_v \nabla^2 v \ + \ u\,v^2 \ - \ (f+k)\,v. \end{aligned} \tag{5}$$

Here, D_u and D_v represent the diffusion constants of the two chemicals, f controls the rate at which U is supplied and $f + k$ controls the rate that V decays.

II. Method

A. Diffusion Limited Aggregation

To implement the DLA into a computational model, we needed to make several design choices, which will be discussed in this section. Subsequently, we will explain which test we did on it and why they are interesting.

The diffusion present in the system is implemented with the SOR iterative method. This method has been shown in previous research to be the most efficient method for a good ω[2]. In order to guarantee the periodicity of the system, it was chosen that all values in column zero are equal to the values in column n, so $n-1$ is used as its neighbor. It is also important to know that the test and figures on the DLA are performed with a tolerance of 10^{-5}, which means that iteration continues until the difference in the matrix is less than that value.

Aggregation is implemented by keeping a list of aggregated cells next to the concentration values. This list has 3 possible states: zeros lay outside the cluster, ones are part of the cluster and twos are candidates for aggregation, neighbouring the cluster. The neighborhood is determined by a Von Neumann neighborhood. When a cell is part of the aggregate, it is assumed that it's concentration will be 0 for ever, meaning that concentration of its neighbour will disappear in it.

The whole system follows this algorithm to add a cell to the cluster at each new step:

1) First, a new cell is chosen from the possible neighbours.
2) After this, the matrix is updated with this value and the new neighbours are determined.
3) With this new matrix, SOR iterations are done until the tollerance is reached: *the diffusion is "solved"*. After which the loop is executed again.

The test on this system will consist of two parts, first we will visually investigate the influence of η after which we will also study the concentration distribution of 100 cell additions to the aggregation for different values. After

this we will study with three different η and see if there is an optimal ω for the SOR with DLA.

In addition to this, we also researched ways to improve on SOR by solving the grid in alternative ways. We implemented the *Multigrid* approach where we solved the domain with SOR over a coarse space grid, we then applied the results to the underlying fine grid steps, and finally solve the finer grid. The cluster was handled by considering any coarser grid step as part of an overlying cluster if *any* underlying fine grid step was part of the finer cluster.

B. Monte-Carlo Simulation

Within the same domain in which we simulated our DLA model until now, we will attempt to reproduce the same objects using Monte Carlo (MC) (I-B).

1) Random Sampling: We model our DLA system with a seemingly more grounded way to simulate particle diffusion: random walk. One-by-one, we generate particles at the top of our domain and have them move horizontally and vertically at random. When stretched to ∞ this process converges towards diffusion motion. The particles move randomly, yet respect some core rules:

- If a particle crosses the left and right periodic boundaries, they reappear on the other side of the domain.
- If one crosses the bottom or top boundaries it permanently leaves the domain; it disappears.
- Predominantly, if a particle comes in the vicinity of the cluster it may stick/**aggregate** to it.

We generated particles until the cluster was 400 particles strong.

2) Aggregation: Initially 100% of particles in contact with the cluster aggregated to it. We later tested the model when the particle have a chance to keep wondering in the available free space for some probability $p = 1 - p_s$. p_s is the probability for a free particle to stick to the cluster if next to it.

Just like the parameter η had influence on the probability to aggregate/growth which resulted in a change in the cluster's shape, we tested how varying p_s affected it as well.

3) Density vs Aggregation probability: Monte Carlo yields clusters which are more or less dense; from one run to the next, but also -seemingly- depending on p_s. We tested a likely correlation between the two by qualifying a cluster's density with the area of the smallest rectangle (parallel to the domain) in which the cluster (of 400 particles always) fits.

For each of 9 values of $0.1 \leq p_s \leq 0.9$, we produced 100 clusters and their average density.

Number	D_u	D_v	f	k
1	0.16	0.08	0.035	0.060
2	0.14	0.06	0.025	0.055
3	0.16	0.08	0.025	0.050
4	0.18	0.09	0.040	0.065
5	0.16	0.08	0.045	0.070

TABLE I: Selected parameters

C. Gray-Scott Model

The model was implemented in two dimensions, using the equations seen previously. To solve these equations numerically, we discretize them using a forward Euler method:

$$\frac{\partial u}{\partial t} \approx \frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t}. \tag{6}$$

This results in the following discretizations:

$$\begin{aligned} u_{i,j}^{n+1} &= u_{i,j}^{n} + \Delta t \left(D_u \nabla^2 u - uv^2 + f(1-u) \right), \\ v_{i,j}^{n+1} &= v_{i,j}^{n} + \Delta t \left(D_v \nabla^2 v + uv^2 - (f+k)v \right). \end{aligned} \tag{7}$$

To speed up the program, just-in-time (JIT) compilation was used via Python package *Numba*.

The system was initialised with a 100×100 grid with $U = 0.5$ everywhere, $V = 0.5$ in a small square in the centre, and $V = 0$ elsewhere. Noise of 0.05 was added to both U and V. $\delta t = \delta x = 1$ and was also used. Each simulation ran for 10000 timesteps, with snapshots taken at $t = 0, 1000, 2500, 5000, 7500, 10000$. However, for space-saving reasons, the first and last snapshots are not part of the plots included in this paper.

Simulations were made for five different parameter sets, which can be seen in Table I.

The first parameter set was chosen as a baseline for the simulation. The second set used slightly lower D_u, D_v, f, k values, which may lead to slower diffusion and more stable patterns. The third set kept the same diffusion rates but reduced the reaction rates, potentially leading to longer-lasting patterns. The fourth set explored slightly increasing all parameters, likely leading to less stability. Finally, the fifth set experimented with the same diffusion rates but higher reaction rates. This may accelerate the system's transitions.

III. Results

A. Diffusion Limited Aggregation

As discussed earlier, the evaluation of the DLA system consists of three different tests. In the first test, we analyze the multi-plot in Figure 1, in which the clusters formed for different values of η are visually compared.

The figure shows that for higher values of η the probability of growing upwards seems to increase, resulting in a straighter vertical structure. In contrast, a value of $\eta < 1$ results in a broader and flatter structure. When $\eta = 1$, the distribution seems more balanced, with a shape that lies between the two extremes. Good to note is that the red values in the right column are a result of interpolation of the package used to visualize. As all the values are either 0 or 1.

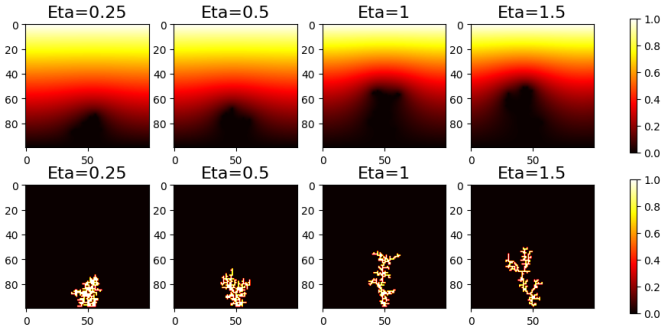

Fig. 1: Visualization of DLA for different Eta values. With 250 aggregations.

Figure 2 shows that the trends observed in Figure 1 are also reflected in the concentration plot. It is clear when the aggregation growhs higher, its influence on the entire system becomes bigger. This comes due to the fact that the sinkholes are taking more concentration from the system.

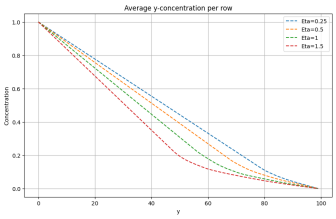

Fig. 2: Average concentration per row

Figure 3 shows how the value of ω affects the number of iterations needed to arrive at a final result for 100 aggregates. It is clear that for a lower value of η fewer iterations are needed. Furthermore, the average results show that there seems to be an optimal value for ω. However, it is important to note that this does not hold for all tests, since the construction of the simulation is arbitrary. This means that the choice of ω is not always the most efficient one. This is illustrated by the relatively large standard deviation in some parts of the data.

Note. *Our Multi-grid implementation for solving SOR in our system showed a significant improvement. We produced a* 400 *sized cluster over a* 500×500 *grid in*

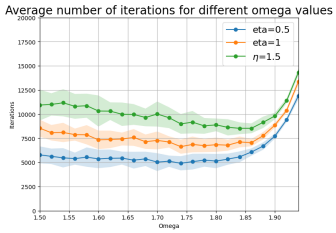

Fig. 3: Average number of iterations for different eta values over 20 runs with 100 aggregations. With the tolerance for the SOR being 10^{-5}. The lighter shading represents the standard deviation, indicating the differences between the runs

16933 iterations with SOR vs 1881 total iterations by first solving the overarching 50×50 *grid, with* $\omega = 1.7$.

B. Monte-Carlo Simulation

Having implemented our Monte Carlo model for solving our system, we show in figure (4) that it produces a similar 400-particle cluster over our 100×100 grid then with our initial model with $\eta \approx 1.7$.

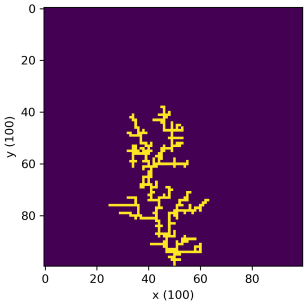

Fig. 4: Monte Carlo Simulation of a cluster of 400 particles

Although we re-counted our clusters from the previous model with $1.4 \leq \eta \leq 2$, we could not manipulate the available parameter to force a more compact cluster like a low η permitted, until with introduction the possibility to fail the aggregation.

We ran the same model for 9 values for p_s (figure 5); the probability that a particle entering the neighborhood of our cluster will stick to it, as opposed to continuing to move around until it aggregates or leaves the domain successfully. The probabilities tested ranged uniformly between 0.1 and 0.9.

We observed a non-definitive compactness for lower values of p_s. We found this intuitive: the cluster usually consists of fractal *bays*: ”branches surrounding an aperture”. The only way we can hope to aggregate a particle

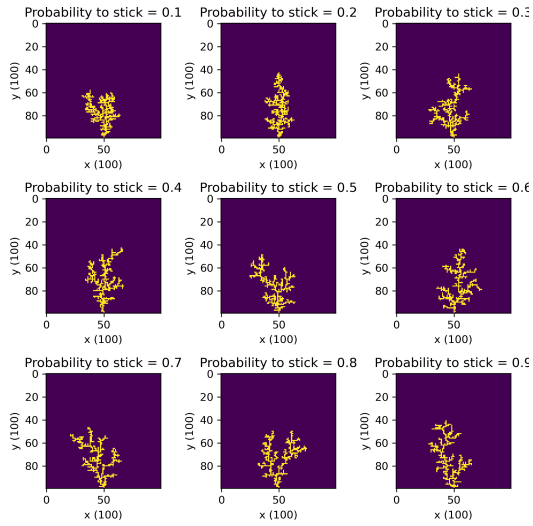

Fig. 5: MC Simulation with Varying Probabilities for Aggregation

within one of those bays, let alone fill it, is when a particle "*enters*" the bay without aggregating. If $p_s = 1$ then this is impossible for any entry narrower than 3 free locations. Intuitively, the lower p_s is, the most likely it will be for a particle to venture past a narrow entry and aggregate within an existing structure of the cluster (fill in gaps). While with high p_s, the only locations available for aggregation are ones in the outskirts of the cluster, forcing the object to grow outward. But such a correlation could not be confirmed without doing many runs as our model is, by this point, highly stochastic.

In order to hopefully confirm this hypothesis, we compared the density (described in II-B3) of samples of 100 clusters for each of the above values for p_s. Figure 6 shows the mean and standard variations of each samples, plotted over the corresponding values of p_s.

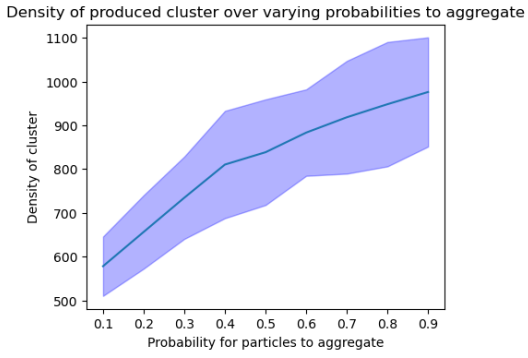

Fig. 6: Density of Clusters with Varying Aggregation Probability

This served to confirm that the parameter p_s, in the case of our MC model, can act (among other things: speed, efficiency, ...) as a shape parameter for our resulting clusters, just like η did within our initial DLA.

Overall, our attempt at simulating a ***Monte Carlo*** *version of our DLA system showed that one can bypass the discretized layering of several reactions to integrate a system with interacting dynamics. We succeeded in creating an analogous model to* ***Diffusion Limited Aggregation*** *governed by a single governing dynamic.*

C. Gray-Scott Model

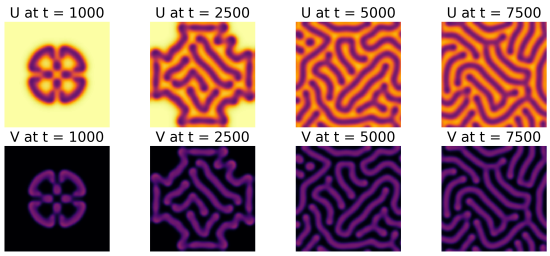

Fig. 7: $D_u = 0.16, D_v = 0.08, f = 0.035, k = 0.06$

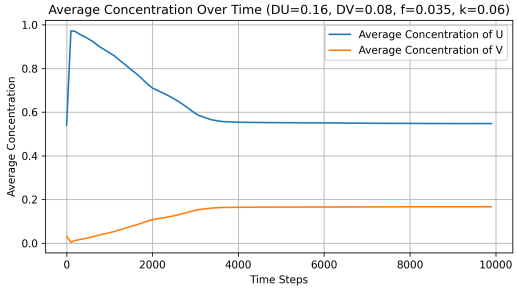

Fig. 8: Average concentration of chemicals over time of $D_u = 0.16, D_v = 0.08, f = 0.035, k = 0.06$

Figure 7 shows some interesting symmetric patterns appearing at $t = 1000$. At $t = 2500$, patterns become more complex, forming Turing structures. At the final three time steps, we see the system stabilize, forming maze patterns. The stability of the two chemicals U and V can be seen in Figure 8, where it seems around $t = 4000$, the system has reached a stable state, and the concentrations don't change much.

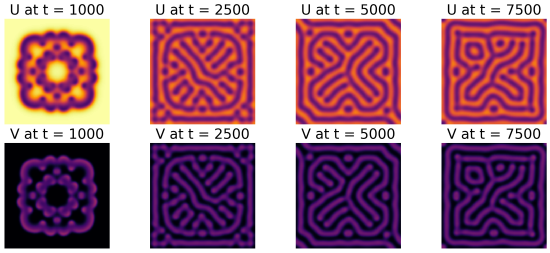

Fig. 9: $D_u = 0.14, D_v = 0.06, f = 0.025, k = 0.055$

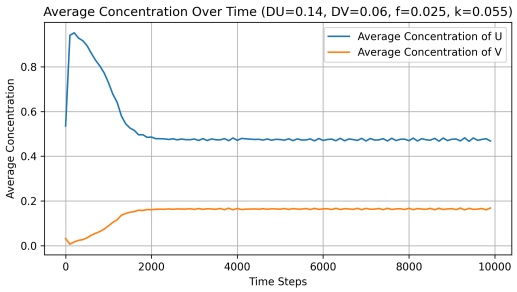

Fig. 10: Average concentration of chemicals over time of $D_u = 0.14, D_v = 0.06, f = 0.025, k = 0.055$

Figure 9 shows the behaviour of the system with all four parameters slightly reduced. We see a more complex pattern forming at $t = 1000$ than the baseline, suggesting that the reaction is more advanced. For the next four time steps recorded, it seems a stable state has already been reached. This can be confirmed in Figure 10, where it seems a stable state has been reached just after $t = 2000$. Interestingly, unlike the baseline, there seem to be small persistent fluctuations in the concentrations of the chemicals after this stable state is reached.

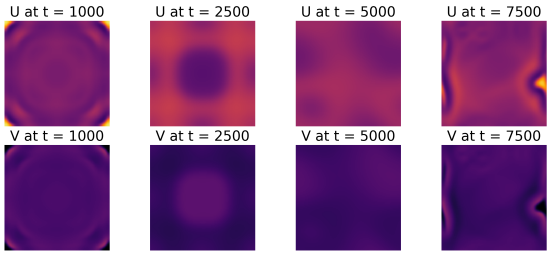

Fig. 11: $D_u = 0.16, D_v = 0.08, f = 0.025, k = 0.05$

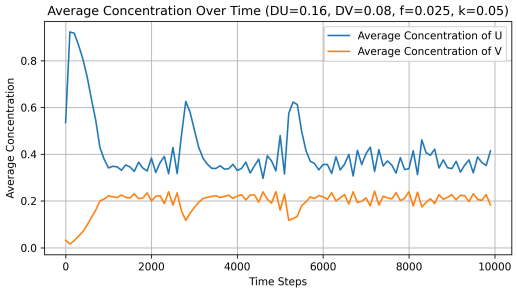

Fig. 12: Average concentration of chemicals over time of $D_u = 0.16, D_v = 0.08, f = 0.025, k = 0.05$

Figure 11 shows the behaviour of the system with the same diffusion parameters but a slightly lower f and k value. The behaviour of this system is very different, as no discernable patter can be noticed, instead, there seems to be waves that appear and expand outwards. We can see in Figure 12 that the system experiences spikes in the concentrations of U and drops in V, with significant ones around $t = 3000$ and $t = 5500$. When the simulation was run until $t = 30000$, it seemed no stable state had been reached yet, and small wave patterns were still visible.

Graphs for the fourth and fifth parameter sets are not included here. Both systems diffused too quickly, and no patterns were created. This was even the case when the graph was plot for every 100 time steps.

Patterns only form when diffusion is neither too high nor too low compared to the reaction parameters. When diffusion rates are too high, we found that local reactions didn't cause patterns to occur, and instead, the entire system was diffused. On the other hand, when reaction rates are lowered too much, patterns can fail to appear and the system will not stabilize. This shows that extreme values cause the system to become unstable or stagnate.

The appearance of persistent fluctuations in some systems is interesting, as though the simulation looked stable visually, the average concentration of U and V proved otherwise.

IV. Discussion and Conclusion

We have modelled Diffusion Limited Aggregation, a system of conflicting, interacting dynamics in with two major approach. When solving the system with an overlap of *time-independent diffusion solving* with Successive over-relaxation method and *aggregating growth*, we stumbled onto similar constraints as within a single dynamic diffusion system, where we were limited by our efficacy for solving the diffusion of the grid. The Monte Carlo approach allowed us to produce an analogous model derived from a *unique* dynamic, albeit stochastic and costly, which limited us in producing definitive results.

We investigated the reaction-diffusion mechanics of the Gray-Scott model by simulating it with various parameter sets. By adjusting these parameters, we observed various behaviours, such as stable Turing and maze patterns, as well as chaotic wave formations and complete diffusion. We found that a delicate balance between diffusion and reaction rates was required in order for patterns and a stable system to occur.

References

[1] Thomas Norton Balthazar Dupuy d'Angeac Ruben Lanjouw. "Scientific Computing Exercise Set 1". In: (2024).

[2] UvA. *Assignments for Scientific Computing*. en. 2021.