

Full Finite Element Scheme for Reaction-Diffusion Systems on Embedded Curved Surface in \mathbb{R}^3

Candidate number:

August 31, 2023

Contents

| | | |
|----------|--|----------|
| 1 | Introduction | 1 |
| 1.1 | Basics of Turing patterns | 1 |
| 1.2 | Significance and applications of Turing patterns on surfaces | 2 |
| 1.3 | The structure of article | 2 |
| 2 | Docker, Gmsh and Firedrake | 3 |
| 2.1 | Introduction to Docker | 3 |
| 2.2 | Introduction to Gmsh | 3 |
| 2.3 | Introduction to Firedrake | 3 |
| 2.4 | Advantage of Docker | 4 |
| 2.5 | Advantage of Gmsh | 4 |
| 2.6 | Advantage of Firedrake | 5 |
| 3 | Formation of Surface Based on Gmsh | 5 |
| 3.1 | Mesh Formation Process | 6 |
| 3.2 | Generate For plane with hole | 6 |
| 3.3 | Generate For Sphere | 6 |
| 3.4 | For Cylinder | 8 |
| 3.5 | For Torus | 9 |
| 4 | Process of Calculation based on Firedrake | 9 |
| 4.1 | Install firedrake based on Docker | 12 |
| 4.2 | Import Mesh to Firedrake | 12 |

| | | |
|----------|--|-----------|
| 4.3 | Weak form of Schnakenberg model | 13 |
| 4.4 | UFL Representation | 14 |
| 4.5 | Forward Euler strategy for numerical solution | 14 |
| 4.6 | Visualization of solution | 14 |
| 5 | Numerical results of Schnakenberg model | 16 |
| 5.1 | Introduction to Schnakenberg model | 16 |
| 5.2 | Mathematical equation of Schnakenberg Model | 16 |
| 5.3 | Strong and Weak form of Schnakenberg model | 17 |
| 5.4 | Numerical Calculation code based on Firedrake | 17 |
| 5.5 | Schnakenberg model for square with flower | 19 |
| 5.6 | Schnakenberg model for square with hole | 19 |
| 5.7 | Schnakenberg model for Sphere | 19 |
| 5.8 | Schnakenberg model for torus | 22 |
| 5.9 | Schnakenberg model for other surface | 22 |
| 5.10 | Discussion of Schnakenberg Model | 27 |
| 6 | Numerical results of Gierer-Meinhardt model | 27 |
| 6.1 | Introduction to Gierer-Meinhardt model | 27 |
| 6.2 | Mathematical equation of Schnakenberg Model | 28 |
| 6.3 | Weak form of Gierer-Meinhardt model | 28 |
| 6.4 | Gierer-Meinhardt for square with hole | 29 |
| 6.5 | Gierer-Meinhardt model for sphere | 30 |
| 6.6 | Gierer-Meinhardt model for torus | 33 |
| 7 | Numerical results of FitzHugh-Nagumo model | 33 |
| 7.1 | Introduction to FitzHugh-Nagumo model | 33 |
| 7.2 | Mathematical equation of FitzHugh-Nagumo Model | 38 |
| 7.3 | Weak form of FitzHugh-Nagumo model | 39 |
| 7.4 | FitzHugh-Nagumo model for square with hole | 40 |
| 7.5 | FitzHugh-Nagumo model for sphere | 40 |
| 7.6 | FitzHugh-Nagumo model for torus | 44 |
| 8 | Compare three models | 44 |
| 8.1 | Similarities | 44 |
| 8.2 | Differences | 46 |

| | |
|--|-----------|
| 9 Conclusion | 46 |
| 9.1 Main contributions of the research | 47 |
| 9.2 Future directions and recommendations for research | 47 |

Abstract

Reaction-diffusion systems are a class of mathematical models used to describe the propagation and interaction of substances or physical phenomena in space, where reactions and diffusion collectively influence the system's evolution and pattern formation. Studying reaction-diffusion systems on curved surfaces offers insights into how intricate geometries influence the dynamics of chemical and biological processes. It opens up opportunities to develop novel applications and technologies that leverage the unique properties of curved surfaces for controlled reactions and diffusion processes. In this thesis, we comprehensively explore the application of the finite element method to solve the Schnakenberg equation, Gierer-Meinhardt model, and FitzHugh-Nagumo model. The study includes tool usage, numerical solving, and parameter analysis. Proficiency in employing advanced software tools like Gmsh, Docker, and Firedrake was showcased. An extensive investigation was carried out into the influence of diverse parameters on the numerical solutions of distinct models at each time step, thereby enhancing the understanding of reaction-diffusion systems. Additionally, perceptive analysis was provided concerning the effects of diffusion coefficients and other parameters on the eventual steady-state solutions on curved surfaces.

1 Introduction

1.1 Basics of Turing patterns

Alan Turing, best known for his pioneering work in the realm of mathematical biology through his study on reaction-diffusion systems[1]. His work delved into the underlying mechanisms by which certain biological patterns form.

At the core of Turing's exploration lies the concept of reaction-diffusion systems[2]. These are mathematical models that describe how concentrations of one or more substances change under the influence of two primary processes: local chemical reactions, where substances are transformed into each other, and diffusion, which causes these substances to spread out over a space. [3, 4]

Turing was interested in morphogenesis, the process by which biological organisms develop their shape and structure. He postulated that under certain conditions, a uniform mixture of chemicals can spontaneously form patterns when the rate of diffusion and the nature of the reactions between the chemicals are just right. These patterns could be in the form of spots, stripes, or spirals, which are reminiscent of many patterns observed in nature, such as the stripes on a zebra or the spots on a leopard. His profound insight was that even in situations where a system seems stable and homogeneously distributed, slight perturbations can lead to the development of complex patterns due to the interplay between reactions and diffusion.

The central mechanism Turing proposed revolves around the idea that two or more interacting chemicals, when diffusing at different rates, can create an instability in an initially uniform state. This instability, given the right conditions, can lead to the emergence of patterns. An essential aspect of these patterns is that they arise without any pre-existing template or gradient – a phenomenon that was quite revolutionary at the time.

He derived the governing equations for reaction diffusion mechanisms in form [5]:

$$\frac{\partial \mathbf{c}}{\partial t} = \mathbf{f}(\mathbf{c}) + D \nabla^2 \mathbf{c}$$

where \mathbf{c} is the vector of morphogen concentrations, \mathbf{f} represents the reaction kinetics and D is the diagonal matrix of positive constant diffusion coefficients. In this article, we mainly concerned with models for two chemical species, $A(\mathbf{r}, t)$ and $B(\mathbf{r}, t)$ say. The equation system is then of the form

$$\begin{aligned}\frac{\partial A}{\partial t} &= F(A, B) + D_A \nabla^2 A, \\ \frac{\partial B}{\partial t} &= G(A, B) + D_B \nabla^2 B,\end{aligned}$$

where F and G are the kinetics, which will always be nonlinear.

1.2 Significance and applications of Turing patterns on surfaces

Over the past decades, Turing patterns have been widely applied in various natural and engineering fields, such as biological morphogenesis, ecological system patterns, chemical reactions, and material science.

Considering the numerical solution of Turing patterns on surfaces, the significance of this research is as follows:

- Biological Morphogenesis: The patterns formed on the skin, feathers, scales, etc., of many organisms (such as spots and stripes) may arise from the operation of the Turing mechanism on surfaces. Researching the numerical solutions of Turing patterns on surfaces helps to better understand how these patterns in nature are formed.
- Engineering and Design Applications: In material design, architecture, art, and other creative domains, the behavior of Turing patterns on surfaces could provide designers and engineers with novel tools and approaches, leading to innovative and captivating designs.
- Deeper Understanding of Basic Science: Numerical solutions of Turing patterns not only offer a quantitative description of the phenomenon but also aid in gaining a deeper understanding of fundamental processes like non-linear dynamics, diffusion, and reactions.

1.3 The structure of article

The following chapters will be organized as follows: Chapter 2 introduces three software tools involved in the full finite element Scheme, including Docker, Gmsh, and Firedrake. Chapter 3 describes how to use Gmsh to create plane with hole, 2D sphere, cylinder, and Torus surfaces. Chapter 4 describes the basic install and usage of Firedrake, demonstrates the transformation of the Schnakenberg equation into Weak form, and explains how to solve the Schnakenberg equation using the finite element scheme. Chapter 5 investigates the visualization results of the numerical solutions on plane with hole, sphere, and torus surface for the Schnakenberg equation. Chapter 6 examines the influence of parameters in the Gierer-Meinhardt model on the numerical solutions on plane with hole, sphere, and torus surfaces. Chapter 7 studies the relationship between steady-state solutions and parameters in the FitzHugh-Nagumo model on plane with hole, sphere, and torus surfaces.

Chapter 8 compares the similarities and differences among different models. Chapter 9 summarizes the entire paper and proposes future research directions.

2 Docker, Gmsh and Firedrake

To solve the turing pattern equation on a surface, we harness the synergy of the following three tools: Docker, Gmsh and Firedrake.

Together, these tools not only allow us to model and mesh complex geometries but also to solve intricate partial differential equations within a stable and consistent environment, ensuring the accuracy and trustworthiness of our results.

2.1 Introduction to Docker

Docker offers an isolated and consistent computational environment, ensuring that all calculations and solutions are reproducible under the exact same settings, regardless of where they are executed. This is paramount for research and collaboration, as it safeguards the reliability and consistency of results.[6]

2.2 Introduction to Gmsh

Through Gmsh, we are able to construct precise geometries of the spherical surface and generate high-quality finite element meshes. This is crucial for subsequent numerical computations and equation solving, as a well-constructed mesh can significantly enhance the accuracy of solutions. [7]

2.3 Introduction to Firedrake

Firedrake is our chosen numerical solver for addressing the GS equation on the sphere. It provides an efficient and adaptable framework for finite element solutions to partial differential equations. Combined with the meshes generated by Gmsh, Firedrake can deliver accurate and reliable numerical solutions.

Together, these tools not only allow us to model and mesh complex geometries but also to solve intricate partial differential equations within a stable and consistent environment, ensuring the accuracy and trustworthiness of our results. [8]

2.4 Advantage of Docker

Leveraging Docker for creating and managing containerized applications brings forth a multitude of compelling advantages.

- Consistency Across Environments: Docker containers encapsulate all dependencies, ensuring that applications run uniformly across different stages, from development to production. This removes the typical "it works on my machine" problem, streamlining the deployment process.
- Isolation and Security: Each Docker container runs in isolation, meaning if one application crashes or has a security breach, it doesn't affect others. This offers a layer of security, as malicious attacks or application failures remain confined to their respective containers.
- Portability and Version Control: Docker images can be shared, versioned, and stored in container registries, allowing teams to easily roll back to previous versions or move applications across different cloud environments or hardware.
- Ecosystem and Integration: Docker boasts a rich ecosystem and integrates seamlessly with many popular CI/CD tools, enhancing the DevOps pipeline. This, coupled with the Docker Hub, offers a repository of pre-built containers that can speed up development cycles.

Docker provides a robust, efficient, and portable platform for application containerization. Its consistent environment replication, efficient resource use, and integrative capabilities make it a top-tier choice for modern software development and deployment.

2.5 Advantage of Gmsh

Utilizing Gmsh to numerically form surfaces has several compelling reasons.

- Geometry Creation: Firstly, you define your spherical surface within gmsh.
- Mesh Generation: gmsh produces a finite element mesh for your spherical model, typically triangulated or tetrahedral, depending on your needs.
- Mesh Export: Once satisfied with the generated mesh, you can export it in a standardized format, such as the .msh format, for utilization in other software.

2.6 Advantage of Firedrake

Utilizing Firedrake to numerically solve Turing patterns on surfaces has several compelling reasons.

- Declarative Programming and Unified Form Language (UFL): UFL provides a high-level platform for declaratively detailing finite element discretizations. When grappling with intricate Turing pattern equations, this approach is especially beneficial for transparently expressing the mathematical model.
- Flexible Meshing and Discretization: Solving PDEs on surfaces, especially problems like Turing patterns, requires precise discretization of the surface. Firedrake supports various mesh types, offering the possibility for intricate discretizations on surfaces. Users can opt for the best discretization strategy suited for their specific problem, optimizing both the accuracy of the numerical solution and computational efficiency.
- Efficient Automated Optimization: Turing pattern equations might involve intricate nonlinear terms and derivative computations. Firedrake's automated optimization tools, like COFFEE and TSFC, can handle these tasks autonomously, ensuring efficient and accurate numerical computations.
- Extensibility and Modularity: As further research and developments in Turing pattern models unfold, there might be a need to incorporate new functionalities or models. Firedrake's modular design facilitates such extensions with relative ease.

Firedrake offers an efficient, flexible, and accurate platform for solving Turing patterns on surfaces. Its declarative programming, automated optimization, and robust parallel processing capabilities make it an ideal choice for such problems.

3 Formation of Surface Based on Gmsh

Gmsh is an open-source 3D finite element mesh generator that reads geometric descriptions from `.geo` files and internally carries out 3D discretization to generate finite element meshes. The entire process encompasses the following steps: First, Gmsh parses geometric entities and related modeling commands from the `.geo` file. Based on these geometric entities, Gmsh then initializes mesh sizes and discretizes points, lines, surfaces, and volumes. After this, Gmsh might also optimize the mesh to meet user-specified quality criteria. Ultimately, the generated mesh structure is saved in the `.msh` file format, which can be read and utilized by firedrake and other finite element analysis software.

3.1 Mesh Formation Process

The .geo file can be made through the following process:

1. Defining Vertices: Firstly, we must define the vertices that constitute both the square and its hole.
2. Connecting Edges: Using the defined vertices, create the edges for the external square and the internal hole.
3. Surface Generation: Once all edges are connected, we can define two planes: one for the external square and another for the internal hole.
4. Meshing: Finally, the square surface with a hole can be meshed, preparing it for finite element analysis.

3.2 Generate For plane with hole

As depicted in Fig 1. First, as step 1 of plane with hole, define vertices, Point (1) through Point (4) moving clockwise, specify the rectangle's corners. The circle center is denoted by Point (5), while Point (6) through Point (9) moving counterclockwise.

Second, as step 2 of plane with hole, connecting edges, edges Line (1) to Line (4) mark the rectangle's boundaries, and arcs from Circle (5) to Circle (8) represent the quarter-circle segments of the circular void, centralized at Point (5).

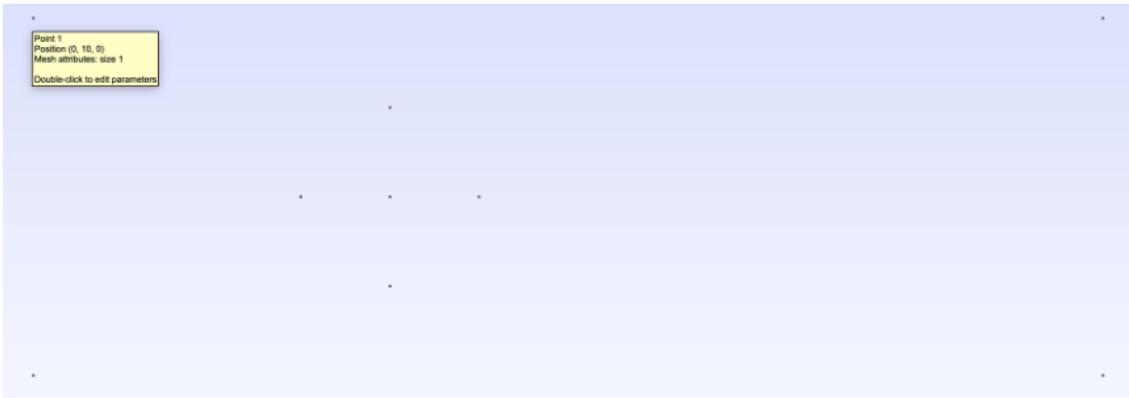
Third, as step 3 of plane with hole, generate surface, the region is further encapsulated by Line Loop (16), a closed structure formed by these line segments, representing the rectangle and its circular aperture. The defined region is then depicted as Plane Surface (16).

Finally, physical entities are demarcated for each segment, labeled from 1 to 5, where Physical Line (4) specifically refers to the quarter-circle arcs of the hole, and Physical Surface (17) symbolizes the entire rectangular region with the circular orifice.

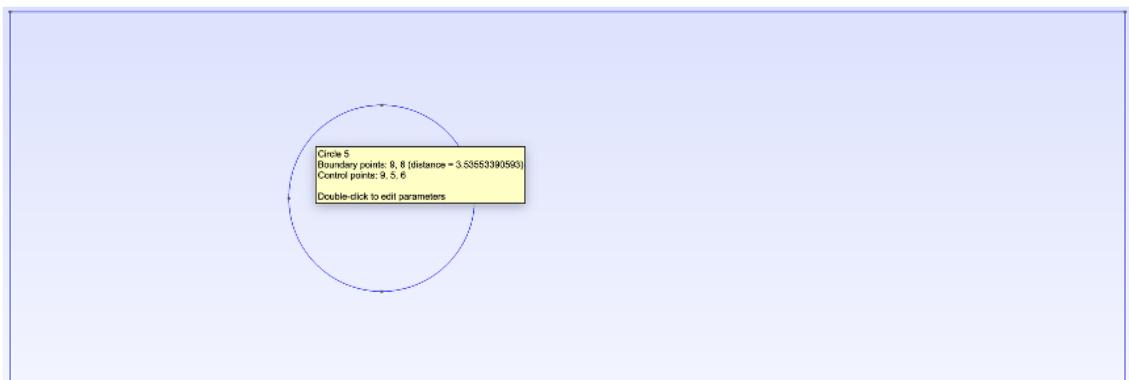
3.3 Generate For Sphere

As depicted in Fig 2. First, as step 1 of sphere, define vertices, Point (1) denote the circle center. Point (2) through Point (5) is the counterclockwise point along the circle moving counterclockwise, while Point (6) through Point (7) indicates the pole of sphere.

Step 1: Define Vertices



Step 2: Define Lines and Circles



Step 3: Define Surface and Mesh

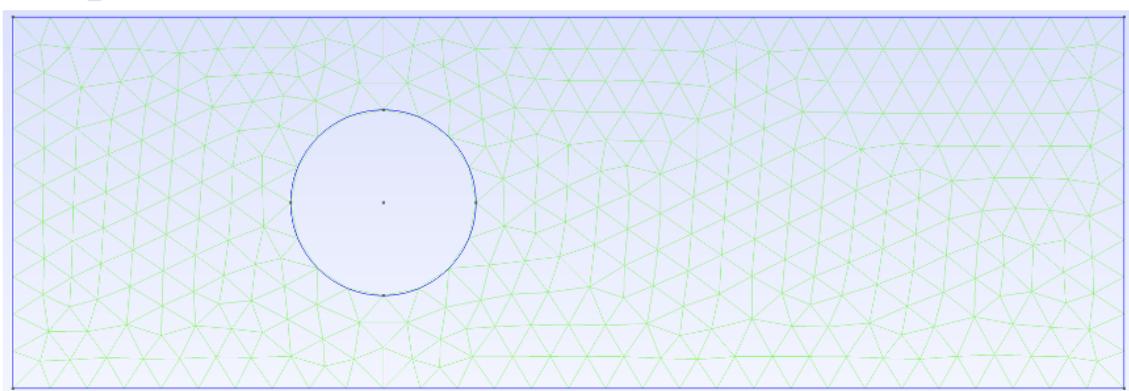


Figure 1: Process of formation of square with hole contain three steps. First, define the vertices of Point (1) to Point (9). Second, connecting edges to generate boundary and circles. Third, define the surface and mesh the surface.

Second, as step 2 of sphere, connecting circles, arcs Circle(1) to Circle(4) mark the circle's boundaries, and arcs from Circle(5) to Circle(8) represent arc to the north pole, while arcs from Circle(9) to Circle(12) represent arc to the south pole.

Third, as step 3 of sphere, generate surface, the region is further encapsulated by Line Loop (16) through Line Loop (21) a closed structure formed by these arc arcs, representing the eight surface of the sphere. The defined region is then depicted as Surface Loop (22).

Finally, physical surface are demarcated for each ruled surface, labeled from 1 to 8 and physical volume is the combination of all ruled surface.

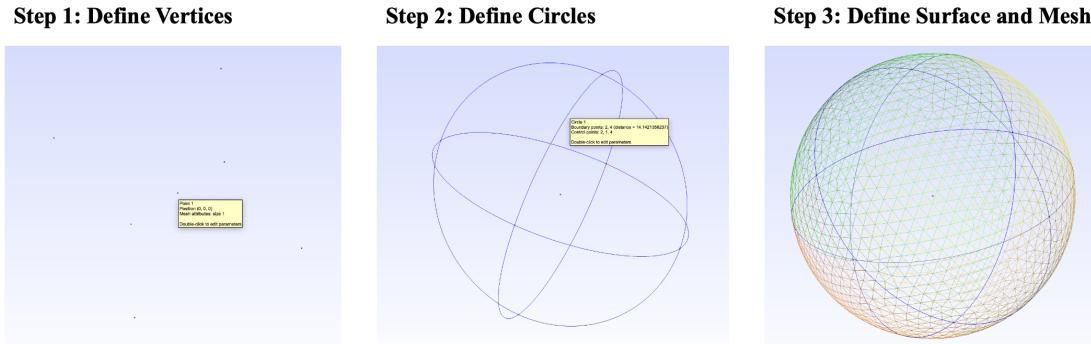


Figure 2: Process of formation of sphere contain three steps. First, define the vertices of Point (1) to Point (7) contain the center, boundary and pole of sphere. Second, connecting arcs to generate circles of each circles. Third, define the surface and mesh the surface.

3.4 For Cylinder

As shown in Fig.3. First, as step 1 of cylinder, define vertices, Point (1) and Point (6) denote the circle center of upper and lower circle respectively. Point (2) through Point (5) is the counterclockwise point along the upper circle moving counterclockwise, while Point (7) through Point (10) indicates the counterclockwise point along the lower circle moving counterclockwise.

Second, as step 2 of cylidner, connecting circles, arcs Circle(1) to Circle(4) mark the upper circle's boundaries, and arcs from Circle(5) to Circle(8) repre-

sent lower circle's boundaries. Circle (9) to Circle (12) represent straight line that connect the upper and lower circle.

Third, as step 3 of cylinder, generate surface, the region is further encapsulated by Line Loop (13) through Line Loop (16) a closed structure representing the four side surface of the cylinder. The upper and lower circle region is then depicted as Surface Loop (17) and Surface Loop (18).

Finally, physical surface are demarcated for each ruled surface, labeled from 1 to 6 and physical volume is the combination of all ruled surface.

In this case, the boundary of the cylinder exhibits high gradients and is completely vertical throughout the entire circumference. The numerical solution is more likely to diverge for a cylinder. Therefore, we choose not to test the numerical solution of the equation on the surface of the cylinder.

3.5 For Torus

As shown in Fig.4. First, as step 1 of torus, define vertices and line, Point (1) and Point (2) denote the two endpoint of the segment and connect these two point to get the segment.

Secondly, as step 2 of torus, obtain the ring by following the line. Proceed by extruding the arc $\pi/2$ in a counterclockwise direction, starting from the zero point, and repeat this process four times. It is important to bear in mind that the inner radius and the outer radius of the ring are determined by the relative distance between the two nodes and the circular shape.

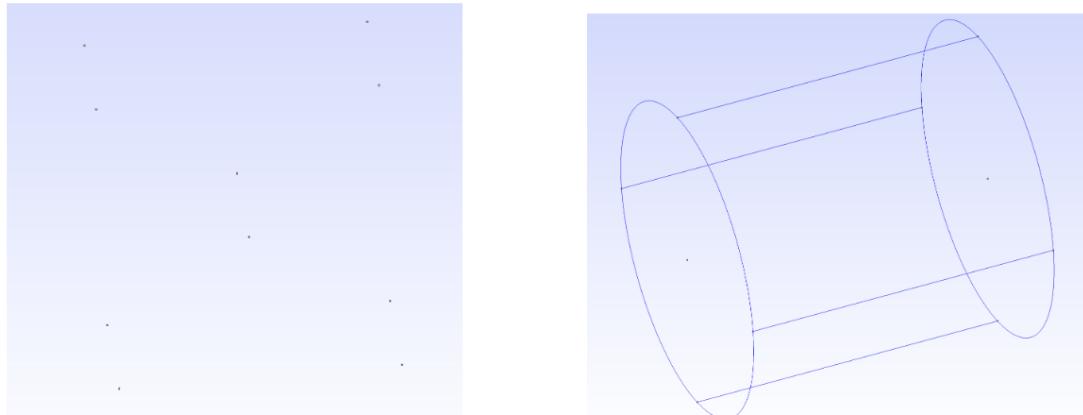
Third, as step 3 of torus, generate torus, extrude the ring $\pi/2$ in a counterclockwise direction, starting from the zero point, and repeat this process four times. Each Time, gmsh leaves the ring insider the torus, so there are totally four ring in torus.

Finally, as step 4 of torus, mesh the torus, physical surface are demarcated for each ruled surface and get the mesh version of torus.

4 Process of Calculation based on Firedrake

In numerical simulations, especially those involving PDEs like reaction-diffusion systems, the quality of the mesh governs both the accuracy and computational cost of the simulation. The surface over which the Turing patterns form needs to be discretized into smaller elements to numerically approximate the solution.

Step 1: Define Vertices of two circles Step 2: Connect Lines and Arcs



Step 3: Mesh the Surface of Cylinder

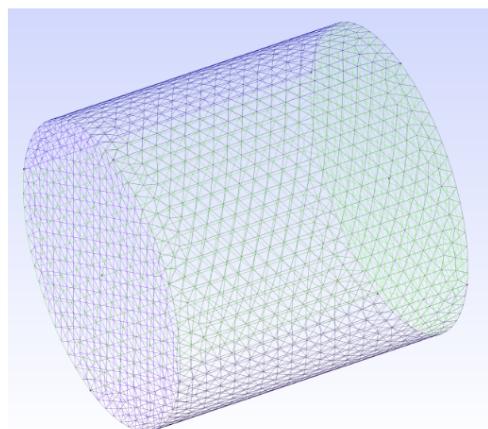
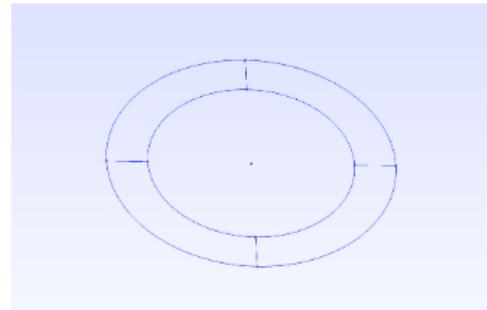


Figure 3: Process of formation of cylinder contain three steps. First, define the vertices of Point (1) to Point (10) contain the center, boundary of circles. Second, connecting arcs and lines to generate circles and connect between circles. Third, define the physical surface and mesh the surface.

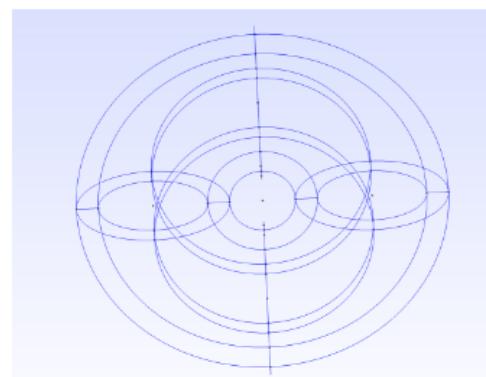
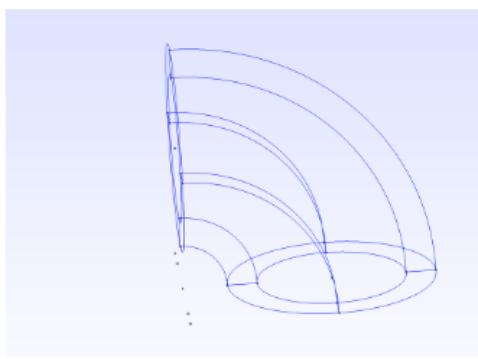
Step 1: Define Vertices and Line



Step 2: Rotate Line to become Ring



Step 3: Rotate Ring to become Torus



Step 4: Mesh Surface of Torus

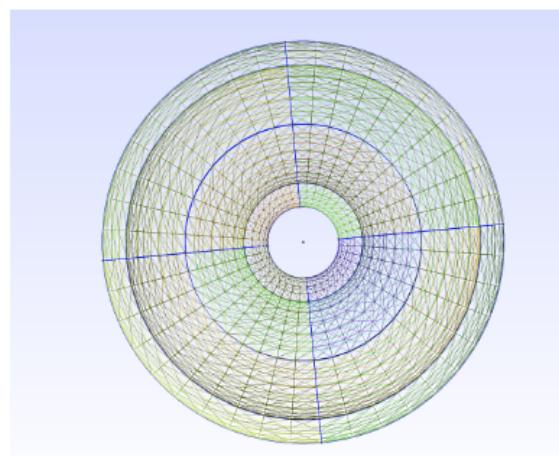


Figure 4: Process of formation of torus contain four steps. First, define the vertices of Point (1) to Point (2) represents the endpoint of segment. Second, rotate the line to become ring. Third, rotate the ring to become torus. Finally, define the physical surface and mesh the surface.

Gmsh is a popular open-source mesh generation software. It allows for the creation of 2D and 3D meshes and offers a rich scripting language. For surfaces with complex geometries, Gmsh can be particularly handy.

First, design the surface geometry using Gmsh's graphical interface or its scripting language. Second, refine the mesh to capture fine details of the geometry. Third, Export the mesh in a .msh format or other formats compatible with Firedrake.

4.1 Install firedrake based on Docker

The installation of the image of Firedrake framework and Jupyter notebook was accomplished by executing the following command:

```
1 docker pull firedrakeproject/firedrake-notebooks
```

Besides downloading the image, we also need to set the file and export the port 8888 for jupyter by executing the following command:

```
1 docker run image_id -p 8888:8888 --volume=/Users/qiyu/docker  
/firedrake:/home/firedrake/file
```

where it build the container, map the port and enable file sharing.

- **Image Building:** Docker run is the command to run a docker contrainer based on the image id.
- **Port Mapping:** -p 8888:8888 maps the host port 8888 to the container port 8888. This allows us to access the container's service running on port 8888 from our host machine.
- **File Sharing:** --volume=/Users/qiyu/:/home/firedrake/ mounts the local directory /Users/qiyu/ to the container directory /home/firedrake/. This enables file sharing between the host and the container.

4.2 Import Mesh to Firedrake

The external mesh can be imported into Firedrake. Here, we are able to import the square with hole, sphere, cylinder and torus.

```
1 from firedrake import Mesh  
2 mesh = Mesh('surface.msh')
```

4.3 Weak form of Schnakenberg model

UFL allows users to focus on the mathematical formulation of their problem rather than the nitty-gritty of its computational implementation. This makes the code more readable and aligned with the mathematical notation used in scientific literature. The abstract representation in UFL can be easily translated to efficient low-level code for various backends and architectures, enabling high-performance simulations without compromising code clarity.

The effects of a reaction-diffusion system on sphere of different radii. We consider the Schnakenberg model, given by

$$\begin{cases} \frac{\partial u}{\partial t} = D_u \nabla^2 u + a - u + u^2 v \\ \frac{\partial v}{\partial t} = D_v \nabla^2 v + b - u^2 v \end{cases}$$

where D_u and D_v are the diffusion coefficients of species u and v , respectively. The numerical results show spot patterns for different cases.

Using UFL in Firedrake, we first need to transfer the formula into weak form by integral. For the first equation, multiply by a test function ϕ , then integrate over the domain Ω :

$$\int_{\Omega} \phi \frac{\partial u}{\partial t} dx = \int_{\Omega} \phi D_u \nabla^2 u dx + \int_{\Omega} \phi (a - u + u^2 v) dx$$

Using integration by parts for the Laplacian term:

$$\int_{\Omega} \phi D_u \nabla^2 u dx = - \int_{\Omega} D_u \nabla \phi \cdot \nabla u dx$$

Similarly, for the second equation, multiply by a test function ψ and integrate:

$$\int_{\Omega} \psi \frac{\partial v}{\partial t} dx = \int_{\Omega} \psi D_v \nabla^2 v dx + \int_{\Omega} \psi (b - u^2 v) dx$$

Using integration by parts for the Laplacian term:

$$\int_{\Omega} \psi D_v \nabla^2 v dx = - \int_{\Omega} D_v \nabla \psi \cdot \nabla v dx$$

Finally, we achieve that

$$\begin{cases} \int_{\Omega} \phi \frac{\partial u}{\partial t} dx + \int_{\Omega} D_u \nabla \phi \cdot \nabla u dx - \int_{\Omega} \phi (a - u + u^2 v) dx = 0 \\ \int_{\Omega} \psi \frac{\partial v}{\partial t} dx + \int_{\Omega} D_v \nabla \psi \cdot \nabla v dx - \int_{\Omega} \psi (b - u^2 v) dx = 0 \end{cases}$$

And further by define the ϕ and ψ test function. We are able to form and solve the solution based on dt and dx :

$$\begin{cases} \int_{\Omega} \phi \frac{u^n - u^{n-1}}{\Delta t} dx + \int_{\Omega} D_u \nabla \phi \cdot \nabla u dx - \int_{\Omega} \phi (a - u + u^2 v) dx = 0 \\ \int_{\Omega} \psi \frac{v^n - v^{n-1}}{\Delta t} dx + \int_{\Omega} D_v \nabla \psi \cdot \nabla v dx - \int_{\Omega} \psi (b - u^2 v) dx = 0 \end{cases}$$

4.4 UFL Representation

Assuming appropriate function spaces V and W have been defined for u and v and assign the value of constant.

```

1 u, v = TrialFunction(V), TestFunction(V)
2 phi, psi = TestFunction(V), TestFunction(V)
3 D_u, D_v, a, b = Constant(D_u_value), Constant(D_v_value),
    Constant(a_value), Constant(b_value)
```

Besides define the parameters, we also need to transfer the weak form above into UFL Representation

$$\begin{cases} \int_{\Omega} \phi (u^n - u^{n-1}) dx + \int_{\Omega} \Delta t D_u \nabla \phi \cdot \nabla u dx - \int_{\Omega} \Delta t \phi (a - u + u^2 v) dx = 0 \\ \int_{\Omega} \psi (v^n - v^{n-1}) dx + \int_{\Omega} \Delta t D_v \nabla \psi \cdot \nabla v dx - \int_{\Omega} \Delta t \psi (b - u^2 v) dx = 0 \end{cases}$$

```

1 F_u = (u - u0)*phi*dx + dt*(D_u*dot(grad(u), grad(phi)))*dx -
    a*phi*dx + u*phi*dx - u*u*v*phi*dx
2 F_v = (v - v0)*psi*dx + dt*(D_v*dot(grad(v), grad(psi)))*dx -
    b*psi*dx + u*u*v*psi*dx)
```

4.5 Forward Euler strategy for numerical solution

Subsequently, we will utilize a Forward Euler strategy [9] to tackle these equations. It entails solving the equations at each time step and subsequently updating the solution values. Note that we need to first assign the initial value of u_0 and v_0 . In each steps. We solve the new value of u, v and replace u_0 and v_0 by the new step value.

```

1 for step in range(num_steps):
2     current_time = step * dt
3     solve(F_u == 0, u)
4     solve(F_v == 0, v)
5     u0.assign(u)
6     v0.assign(v)
```

4.6 Visualization of solution

To visualize Turing patterns on a surface using the Firedrake.

For plane with hole, Firedrake internally leverages an interface to Paraview, which can offer immediate three-dimensional visualization. For instance, assuming you've obtained a solution on some surface.

For sphere, cylinder and torus, we choose Plotly for 3D visualization. First, it provides interactive features that allow users to explore and interact with 3D plots. Second, Plotly produces high-quality, publication-ready visualizations with customizable options for colors, annotations, and styling. Third, Plotly supports a wide range of 3D plot types, including surface plots, scatter plots, mesh plots. Here is the code for plotly for 3D visualization:

```
1 import meshio
2 mesh = meshio.read("surface.msh")
3 points = mesh.points
4 triangles = mesh.cells_dict["triangle"]
5 import plotly.graph_objects as go
6 fig = go.Figure()
7 values = u.dat.data_ro
8 fig = go.Figure(data=[
9     go.Mesh3d(
10         x=points[:, 0],
11         y=points[:, 1],
12         z=points[:, 2],
13         i=triangles[:, 0],
14         j=triangles[:, 1],
15         k=triangles[:, 2],
16         intensity=values,
17         colorscale='Icefire'
18     )
19 ])
```

This will pop up a window displaying the surface and the pattern. The view can be rotated, zoomed, and panned to better inspect the pattern.

5 Numerical results of Schnakenberg model

5.1 Introduction to Schnakenberg model

The study of Turing patterns has garnered attention not only for its insights into the complexities of the natural world but also for its multifaceted applications in areas like materials science, biomedical engineering, and environmental science. While extensive research has been conducted on Turing patterns in unconstrained planes or volumetric spaces, relatively fewer studies have explored systems subject to physical or geometric limitations, such as Turing patterns forming on a plane with a removed section.

In this context, the Schnakenberg model emerges as an invaluable reaction-diffusion system. Initially formulated to describe autocatalytic chemical reactions, this model has evolved to become a versatile tool for studying diverse phenomena including animal coat patterns, cellular organization, and even spatial structures in ecosystems.

In this chapter, we would like to investigate the visualization results of the numerical solutions on plane with hole, sphere, and torus surface for the Schnakenberg equation.

5.2 Mathematical equation of Schnakenberg Model

Mathematically, the Schnakenberg model is described by a pair of partial differential equations:

$$\begin{cases} \frac{\partial u}{\partial t} = D_u \nabla^2 u + a - u + u^2 v \\ \frac{\partial v}{\partial t} = D_v \nabla^2 v + b - u^2 v \end{cases}$$

Here, u and v represent the concentrations of two interacting chemical species, subject to both time and spatial variations. D_u and D_v are the respective diffusion coefficients, while a is a constant source term for u . The parameter b is a rate constant governing the interaction between u and v . The Laplacian operator ∇^2 encapsulates the spatial diffusion effects.

Like the Schnakenberg equation, which has been instrumental in understanding population dynamics and wavefront propagation, the Schnakenberg model serves as a foundational system for exploring the complexities of reaction-diffusion mechanisms and pattern formation. Notably, the model is adept at describing Turing patterns—spatial structures that spontaneously emerge from an initially homogeneous state. This makes it particularly useful for studying the conditions and evolution of pattern formation in constrained geometries.

Therefore, the Schnakenberg model offers a robust framework for both theoretical and practical research, with applications ranging from biological processes to materials science. Its potential to describe complex systems under physical or geometric limitations further extends its applicability, making it a crucial model for ongoing and future studies in various scientific disciplines.

5.3 Strong and Weak form of Schnakenberg model

The Schnakenberg equation of strong form is given by:

$$\begin{cases} \frac{\partial u}{\partial t} = D_u \nabla^2 u + a - u + u^2 v \\ \frac{\partial v}{\partial t} = D_v \nabla^2 v + b - u^2 v \end{cases}$$

Where u represents the concentration or population density that varies with both space and time. D is the diffusion coefficient.

$$\begin{cases} \int_{\Omega} \phi \frac{u^n - u^{n-1}}{\Delta t} dx + \int_{\Omega} D_u \nabla \phi \cdot \nabla u dx - \int_{\Omega} \phi (a - u + u^2 v) dx = 0 \\ \int_{\Omega} \psi \frac{v^n - v^{n-1}}{\Delta t} dx + \int_{\Omega} D_v \nabla \psi \cdot \nabla v dx - \int_{\Omega} \psi (b - u^2 v) dx = 0 \end{cases}$$

```

1 F_u = (u - u0)*phi*dx + dt*(D_u*dot(grad(u), grad(phi))*dx -
   a*phi*dx + u*phi*dx - u*u*v*phi*dx)
2 F_v = (v - v0)*psi*dx + dt*(D_v*dot(grad(v), grad(psi))*dx -
   b*psi*dx + u*u*v*psi*dx)

```

This is the weak form of the Schnakenberg equation. In the finite element method, we seek an approximate solution to this weak form rather than directly solving the original PDE.

5.4 Numerical Calculation code based on Firedrake

We first define the trial and test functions, u and v , which are functions in the function space V . We also define additional test functions ϕ and ψ . Next, we define some constant values D_u , D_v , a , and b . These constants are used in the equations. Then we define the spatial coordinates x and y in terms of the mesh. Next, We interpolate initial expressions $u0_{expr}$ and $v0_{expr}$ into the function space V to obtain initial function values $u0$ and $v0$. We create two Function objects u and v in the function space V to store the current concentrations of u and v . We set the time step size dt to 0.01. Next, we define the variational forms F_u and F_v for the equations governing the evolution of u and v over time. These

forms include terms for the time derivative, diffusion, reaction, and source terms. We set the end time t_{end} to 4.0 and calculate the number of time steps based on the time step size. We define a list output times which specifies the time points at which we want to output the solution. In the time loop, we iterate over each time step. We solve the equations $F_u == 0$ and $F_v == 0$ to obtain the updated values of u and v at the current time step. We then assign these values to $u0$ and $v0$ for the next time step. Finally, we continue the time loop until we reach the final time t_{end} .

```

1 u, v = TrialFunction(V), TestFunction(V)
2 phi, psi = TestFunction(V), TestFunction(V)
3 D_u, D_v, a, b = Constant(1/20), Constant(1), Constant(0.1),
4 Constant(0.9)
5 x, y= SpatialCoordinate(mesh)
6 u0_expr = sin(x*x+y*y)
7 v0_expr = cos(x*x+y*y)
8 u0, v0 = interpolate(u0_expr, V), interpolate(v0_expr, V)
9 u, v = Function(V, name="U-Concentration"), Function(V, name
10 ="V-Concentration")
11 dt = 0.01
12 F_u = (u - u0)*phi*dx - dt*(-D_u*dot(grad(u), grad(phi)))*dx
13     + u*u*v*phi*dx - u*phi*dx + a*phi*dx
14 F_v = (v - v0)*psi*dx - dt*(-D_v*dot(grad(v), grad(psi)))*dx
15     - u*u*v*psi*dx + b*psi*dx
16 t_end = 4.0
17 num_steps = int(t_end / dt)
18 output_times = [0, t_end/4, 2*t_end/4, 3*t_end/4, 4*t_end/4]
19 for step in range(num_steps):
20     current_time = step * dt
21     solve(F_u == 0, u)
22     solve(F_v == 0, v)
23     u0.assign(u)
24     v0.assign(v)
```

5.5 Schnakenberg model for square with flower

In this part, we use Firedrake to solve a partial differential equation (PDE) problem equation and visualizes the results using Matplotlib. We define the constant parameters for the Schnakenberg equation. $D_u = 1/20$ and $D_v = 1$ are the diffusion coefficients, while $a = 0.1$ and $b = 0.9$ are reaction rate parameters. By setting the $u0 = \sin(x * x + y * y)$, $v0 = \cos(x * x + y * y)$, $dt = 0.01$ and $t_{end} = 4$. We use full finite element Scheme for Schnakenberg equation and solve the result of each timestamps of $T = 0, 1, 2, 3, 4$.

As shown in Fig.5, Schnakenberg equation perform classic turing pattern on different times. Our code performs the PDE solution at each time step and visualizes it at specified output times, allowing us to observe how the simulation evolves over time.

5.6 Schnakenberg model for square with hole

As shown in Fig.6, Schnakenberg equation perform classic turing pattern on different times. Our code performs the PDE solution at each time step and visualizes it at specified output times, allowing us to observe how the simulation evolves over time.

In such a plane with a hole, the Schnakenberg model can be used not only to study the conditions for the formation of Turing patterns but also to explore how these patterns evolve over time. Furthermore, the geometry and location of the hole can impact the stability and complexity of the patterns.

5.7 Schnakenberg model for Sphere

The application of the Schnakenberg model on a sphere offers an intriguing and challenging platform for studying pattern formation across various natural phenomena, including biology, chemistry, and physics. On a sphere, Turing patterns may manifest complexities and diversities different from those in a plane or unconstrained three-dimensional space due to geometric constraints and curvature. This provides a valuable research subject for studying how complex patterns form under physical or geometric limitations.

We define the constant parameters for the Schnakenberg equation. $D_u = 1/20$ and $D_v = 1$ are the diffusion coefficients, while $a = 0.1$ and $b = 0.9$ are reaction rate parameters. By setting the $u0 = \sin(x * x + y * y)$, $v0 = \cos(x * x + y * y)$, $dt = 0.01$ and $t_{end} = 3$. We use full finite element Scheme for Schnakenberg equation and solve the result of each timestamps of $T = 0, 1, 2, 3$. As shown in Fig.7, our code performs the PDE solution at each time step and visualizes it at specified output times, allowing us to observe how the simulation evolves over time.

Square with flower

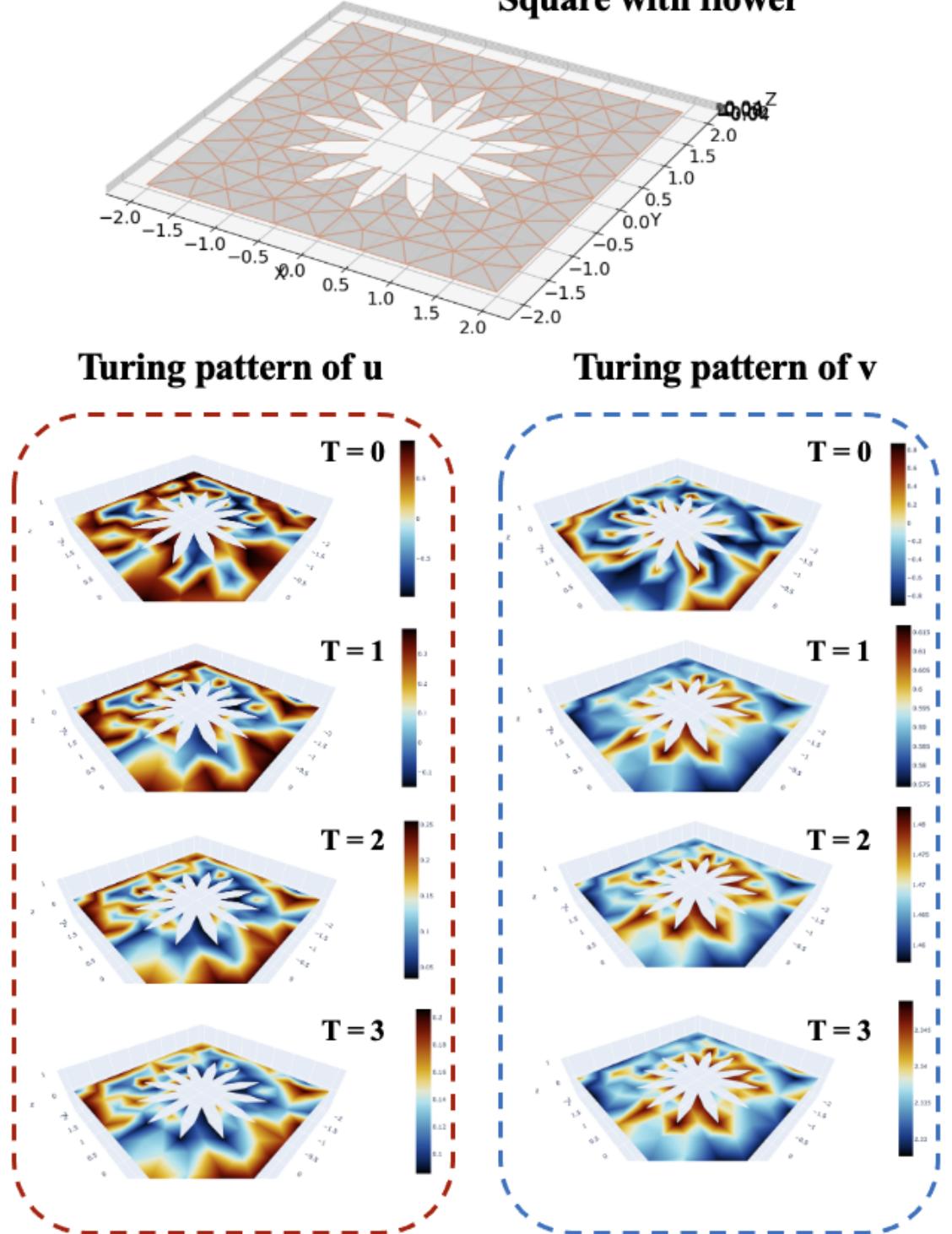


Figure 5: Schnakenberg model for flower. Firstly, we showcase the pattern of the mesh applied to the square with a hole. Each triangle represents an element that we will solve for. We then calculate the equation solutions for u and v for $D_u = 1/20$, $D_v = 1$, $a = 0.1$ and $b = 0.9$ at $T = 0, 1, 2, 3$ respectively.

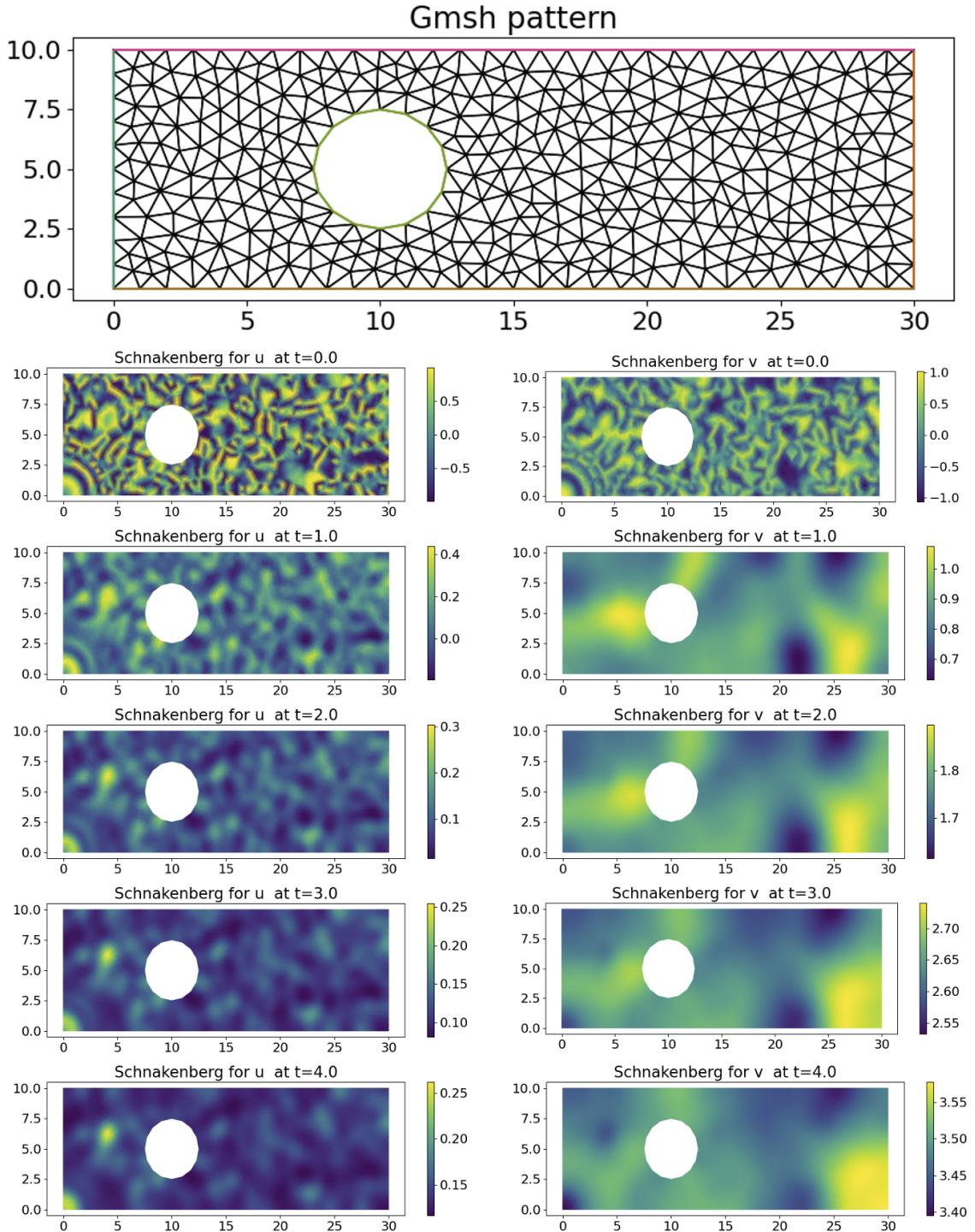


Figure 6: Schnakenberg model for square with hole. Firstly, we showcase the pattern of the mesh applied to the square with a hole. Each triangle represents an element that we will solve for. We then calculate the equation solutions for u and v for $D_u = 1/20$, $D_v = 1$, $a = 0.1$ and $b = 0.9$ at $T = 0, 1, 2, 3, 4$ respectively.

Therefore, the application of the Schnakenberg model on a sphere offers a robust tool for both theoretical research and practical applications, especially in scenarios with physical or geometric constraints. This not only aids in enhancing our understanding of pattern formation mechanisms in the natural world but also presents rich possibilities for future research across multiple scientific disciplines.

5.8 Schnakenberg model for torus

The application of the Schnakenberg model on a torus provides a unique and challenging setting for the study of Turing patterns and reaction-diffusion systems.

We define the constant parameters for the Schnakenberg equation. $D_u = 1/20$ and $D_v = 1$ are the diffusion coefficients, while $a = 0.1$ and $b = 0.9$ are reaction rate parameters. By setting the $u0 = \sin(x * x + y * y)$, $v0 = \cos(x * x + y * y)$, $dt = 0.01$ and $t_{end} = 3$. We use full finite element Scheme for Schnakenberg equation and solve the result of each timestamps of $T = 0, 1, 2, 3$. As shown in Fig.??, our code performs the PDE solution at each time step and visualizes it at specified output times, allowing us to observe how the simulation evolves over time.

Due to the inherent geometric characteristics and curvature of the torus, Turing patterns on this surface may exhibit complexities that differ from those in a plane or in unconstrained three-dimensional spaces. For example, patterns might demonstrate different symmetries or periodicities along different directions of the torus.

5.9 Schnakenberg model for other surface

The application of the Schnakenberg model on a 2d surface provides a study of Turing patterns and reaction-diffusion systems.

First, we define the constant parameters for the Schnakenberg equation. $D_u = 1/20$ and $D_v = 1$ are the diffusion coefficients, while $a = 0.1$ and $b = 0.9$ are reaction rate parameters. By setting the $u0 = \sin(x + y)$, $v0 = \cos(x - y)$, $dt = 0.01$ and $t_{end} = 3$. We use full finite element Scheme for Schnakenberg equation and solve the result of each timestamps of $T = 0, 1, 2, 3$. As shown in Fig.8, our code performs the PDE solution at each time step and visualizes it at specified output times, allowing us to observe how the simulation evolves over time.

Second, we would also like to test the numerical results for cylinder. Define the constant parameters for the Schnakenberg equation. $D_u = 1/20$ and $D_v = 1$ are the diffusion coefficients, while $a = 0.1$ and $b = 0.9$ are reaction rate parameters. By setting the $u0 = \sin(x + y)$, $v0 = \cos(x - y)$, $dt = 0.01$ and $t_{end} = 3$. We use full finite

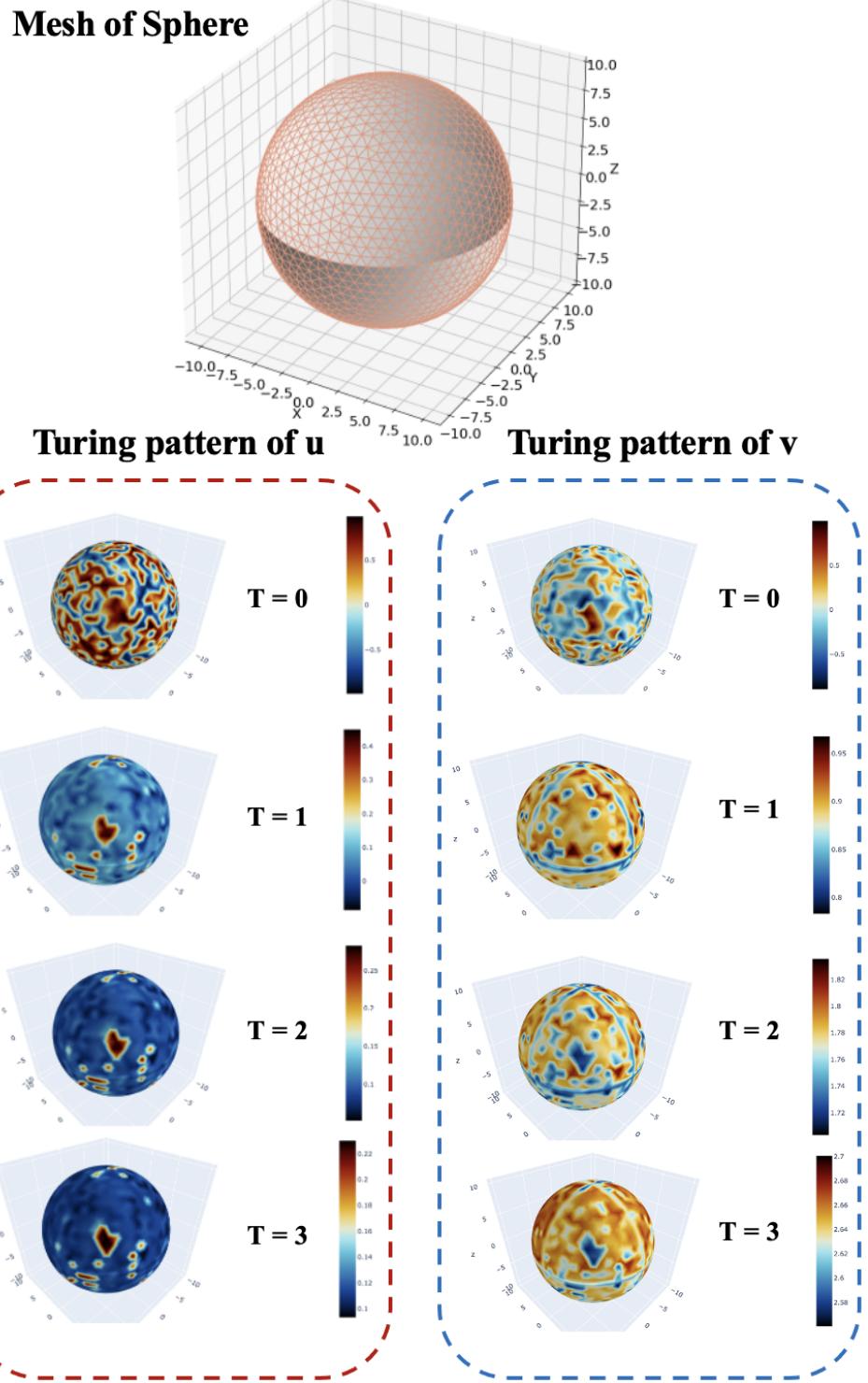


Figure 7: Schnakenberg model for square with hole. Firstly, we showcase the pattern of the mesh applied to the sphere. Each triangle represents an element that we will solve for. We then calculate the equation solutions for u and v for $D_u = 1/20$, $D_v = 1$, $a = 0.1$ and $b = 0.9$ at $T = 0, 1, 2, 3$ respectively.

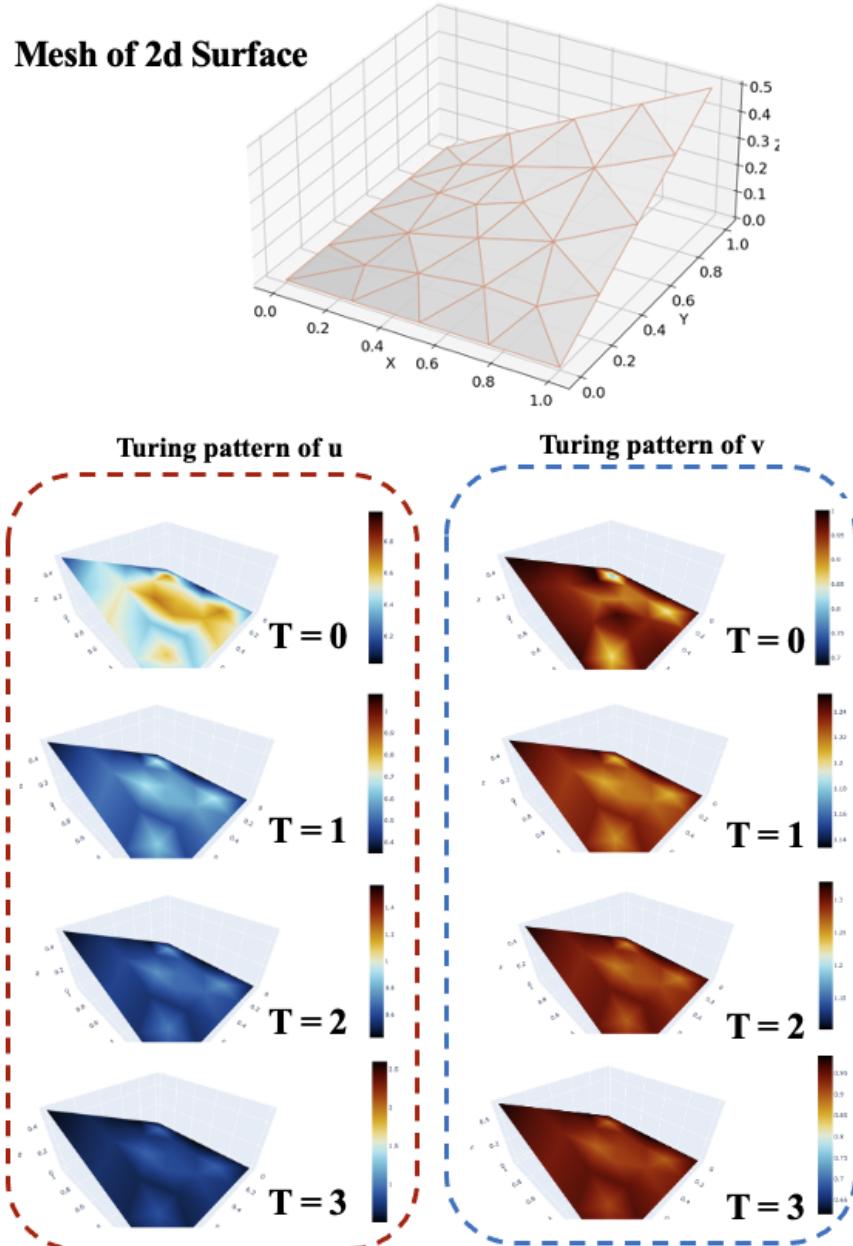


Figure 8: Schnakenberg model for 2d surface. Firstly, we showcase the pattern of the mesh applied to the surface. Each triangle represents an element that we will solve for. We then calculate the equation solutions for u and v for $D_u = 1/20$, $D_v = 1$, $a = 0.1$ and $b = 0.9$ at $T = 0, 1, 2, 3$ respectively.

element Scheme for Schnakenberg equation and solve the result of each timestamps of $T = 0, 1, 2, 3$. As shown in Fig.10, our code performs are not able to converge for $t > 0$. It is due to that cylinder exhibits high gradients and is completely vertical throughout the entire circumference. The numerical solution is more likely to diverge for a cylinder. Therefore, we choose not to test the numerical solution of the equation on the surface of the cylinder. Besides, we also tried best to calculate the whole cylinder surface, the result is also not convergent.

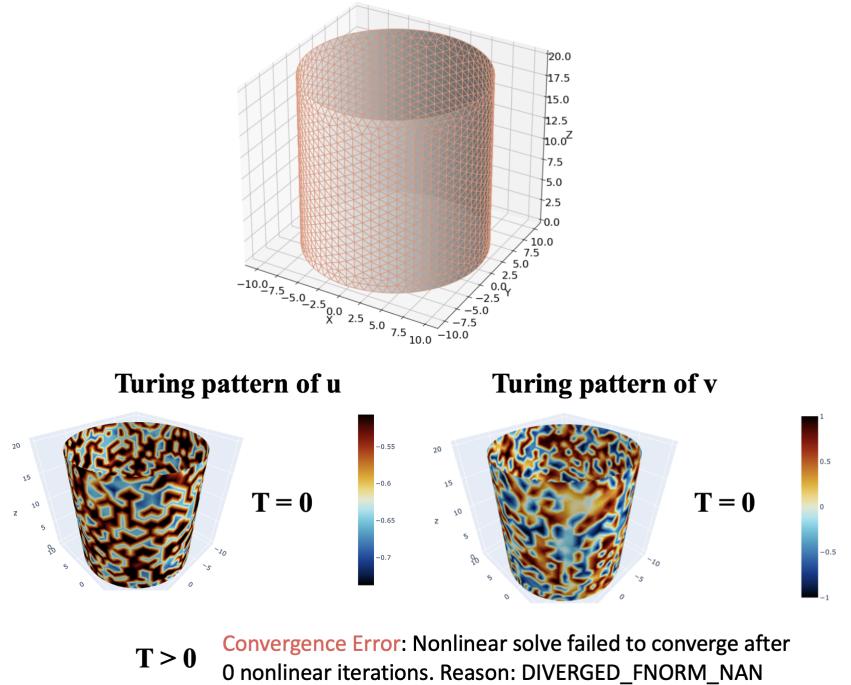


Figure 9: Schnakenberg model for 2d surface. Firstly, we showcase the pattern of the mesh applied to the cylinder. Each triangle represents an element that we will solve for. We then calculate the equation solutions for u and v for $D_u = 1/20$, $D_v = 1$, $a = 0.1$ and $b = 0.9$ at $T = 0$.

Third, we would also like to test the numerical results for cube. Define the constant parameters for the Schnakenberg equation. $D_u = 1/20$ and $D_v = 1$ are the diffusion coefficients, while $a = 0.1$ and $b = 0.9$ are reaction rate parameters. By setting the $u_0 = \sin(x + y)$, $v_0 = \cos(x - y)$, $dt = 0.01$ and $t_{end} = 3$. As shown in Fig.10, our code use full finite element Scheme for Schnakenberg equation and solve the result of each timestamps of $T = 0, 1, 2, 3$.

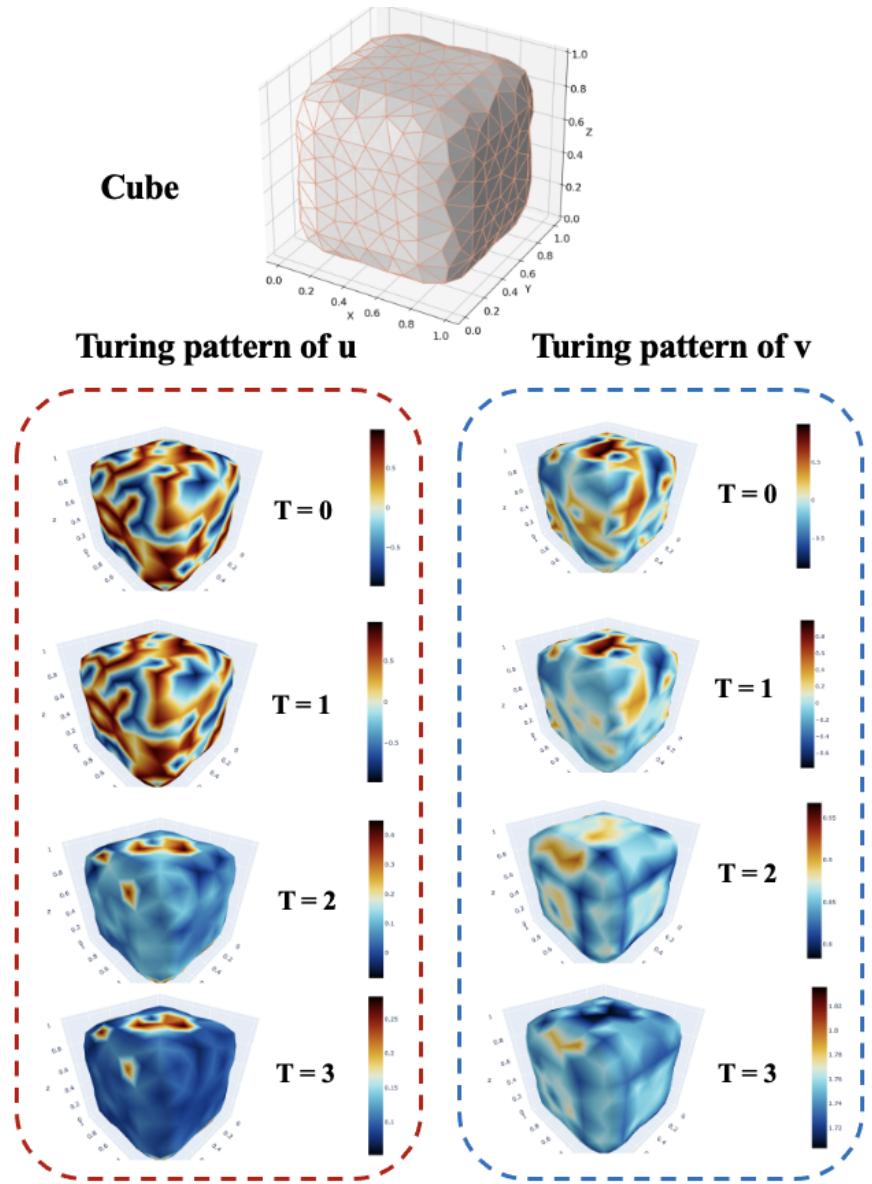


Figure 10: Schnakenberg model for cube. Firstly, we showcase the pattern of the mesh applied to the cylinder. Each triangle represents an element that we will solve for. We then calculate the equation solutions for u and v for $D_u = 1/20$, $D_v = 1$, $a = 0.1$ and $b = 0.9$ at $T = 0$.

5.10 Discussion of Schnakenberg Model

We solve the Schnakenberg model of full finite element on square with hole, sphere and torus. However, the numerical calculation is relatively different for cylinder because the boundary of the cylinder exhibits high gradients and is completely vertical throughout the entire circumference. The numerical solution is more likely to diverge for a cylinder. Therefore, we choose not to test the numerical solution of the equation on the surface of the cylinder.

Here is the discussion of Schnakenberg model based on the numerical solution:

- Characteristics of the Solution: The solution exhibits a distinct pattern of spots within certain regions. This phenomenon could be relevant to certain biological or physical applications of the model, such as pattern formation or material distribution.
- Computational Efficiency and Complexity: In terms of computational efficiency, I found that the computational complexity is directly proportional to both the number of mesh points and the quantity of time divisions. This implies that as the simulation precision increases (i.e., more grid points or smaller time steps), the required computational resources will also correspondingly increase.
- Sensitivity to Parameters: the solution (or the image) diffuses more quickly as the diffusion rate increases. This is intuitively consistent, as a higher diffusion rate would imply that the material will spread more rapidly through the space.
- Error Analysis: It's worth noting that numerical errors accumulate over time. Based on my observations, the upper limit of this error should tend to be on the order of e^t . This is particularly important for long-time simulations, as the accumulation of error could lead to inaccuracies in the final solution.
- Surface Characteristics: the solution's stability varies on different types of surfaces. Specifically, the solution on a torus appears to be more stable than that on a sphere, although no distinct spot patterns were observed on the torus.

6 Numerical results of Gierer-Meinhardt model

6.1 Introduction to Gierer-Meinhardt model

The Gierer-Meinhardt model[10] is a foundational reaction-diffusion system that has been extensively studied for its role in understanding pattern formation, particularly in

biological systems. Proposed by Alfred Gierer and Hans Meinhardt in the early 1970s, this model provides a theoretical framework for explaining how complex spatial patterns can emerge from relatively simple interactions between chemical species. The Gierer-Meinhardt model has had a profound impact on theoretical biology, shedding light on a host of developmental and morphological processes. It has been rigorously analyzed both analytically and numerically to delineate the conditions that produce various kinds of stable or evolving patterns. Furthermore, its adaptability to different geometric contexts and boundary conditions makes it an indispensable tool for interdisciplinary research in biology, mathematics, physics, and beyond.

In this article, we aim to investigate the impact of different constant values by comparing the numerical solutions obtained under varying parameter settings.

6.2 Mathematical equation of Schnakenberg Model

In its specific form, the Gierer-Meinhardt model can be described by the following pair of partial differential equations:

$$\begin{cases} \frac{\partial u}{\partial t} = D_u \nabla^2 u + \rho_u \frac{u^2 v}{1+ku^2} - \mu_u u, \\ \frac{\partial v}{\partial t} = D_v \nabla^2 v - \rho_v \frac{u^2 v}{1+ku^2} + \sigma_v, \end{cases}$$

Here, u and v denote the concentrations of two interacting chemical species. The variables D_u and D_v are their respective diffusion coefficients, while ρ_u and ρ_v represent the rate constants for the production and depletion of these species. The terms μ_u and σ_v serve as decay and source rates, respectively, and k introduces a nonlinearity, modulating the autocatalytic production of u .

The model is particularly interesting because of its nonlinear terms, $\frac{u^2 v}{1+ku^2}$, which introduce autocatalytic effects and saturation. These terms play crucial roles in the formation and stability of patterns. They can generate a range of complex spatial configurations, from spots to stripes, depending on the parameter values and initial conditions.

6.3 Weak form of Gierer-Meinhardt model

Let's consider the Gierer-Meinhardt model given by:

$$\begin{cases} \frac{\partial u}{\partial t} = D_u \nabla^2 u + \rho_u \frac{u^2 v}{1+ku^2} - \mu_u u, \\ \frac{\partial v}{\partial t} = D_v \nabla^2 v - \rho_v \frac{u^2 v}{1+ku^2} + \sigma_v, \end{cases}$$

To express these equations in their weak form using Firedrake and UFL, we first multiply each equation by their corresponding test functions ϕ for u and ψ for v , and then

integrate over the domain Ω .

For the equation of u :

$$\int_{\Omega} \phi \frac{\partial u}{\partial t} dx = \int_{\Omega} \phi D_u \nabla^2 u dx + \int_{\Omega} \phi \left(\rho_u \frac{u^2 v}{1+ku^2} - \mu_u u \right) dx$$

Applying integration by parts on the Laplacian term:

$$\int_{\Omega} \phi D_u \nabla^2 u dx = - \int_{\Omega} D_u \nabla \phi \cdot \nabla u dx$$

Similarly for the equation of v :

$$\int_{\Omega} \psi \frac{\partial v}{\partial t} dx = \int_{\Omega} \psi D_v \nabla^2 v dx + \int_{\Omega} \psi \left(-\rho_v \frac{u^2 v}{1+ku^2} + \sigma_v \right) dx$$

Applying integration by parts:

$$\int_{\Omega} \psi D_v \nabla^2 v dx = - \int_{\Omega} D_v \nabla \psi \cdot \nabla v dx$$

Combining these, we arrive at the weak formulations of the equations:

$$\begin{cases} \int_{\Omega} \phi \frac{\partial u}{\partial t} dx + \int_{\Omega} D_u \nabla \phi \cdot \nabla u dx - \int_{\Omega} \phi \left(\rho_u \frac{u^2 v}{1+ku^2} - \mu_u u \right) dx = 0, \\ \int_{\Omega} \psi \frac{\partial v}{\partial t} dx + \int_{\Omega} D_v \nabla \psi \cdot \nabla v dx - \int_{\Omega} \psi \left(-\rho_v \frac{u^2 v}{1+ku^2} + \sigma_v \right) dx = 0. \end{cases}$$

These weak forms are suitable for numerical approximation using finite element methods, as facilitated by libraries like Firedrake as the code form.

```

1 F_u = (u - u0)*phi*dx - dt*(-D_u*dot(grad(u), grad(phi)))*dx
      + rho*u*u*v/(1+k*u*u)*phi*dx - miuu*u*phi*dx
2 F_v = (v - v0)*psi*dx - dt*(-D_v*dot(grad(v), grad(psi)))*dx
      - rho*v*u*u*v/(1+k*u*u)*psi*dx + miuv*psi*dx

```

6.4 Gierer-Meinhardt for square with hole

The Gierer-Meinhardt model[10] is a well-studied reaction-diffusion system that has provided significant insights into pattern formation, especially in biological systems. While a majority of existing research has focused on Turing patterns in unbounded domains or in regular geometries, fewer studies have considered the effects of more complex geometric constraints, such as square with hole.

First, we investigate the impact of the diffusion coefficient D_v on the Turing patterns, using the Gierer-Meinhardt Schnakenberg model for a square domain with a hole. We focus

on two different values for D_v : 1 and $1/20$. The other parameters remain constant and are set as follows: $D_u = 1/20$, $k = 0$, $\rho_u = 0.18$, $\rho_v = 0.36$, $\mu_u = 0.08$, and $\sigma_v = 0.1$.

We perform numerical simulations and the results are shown in Fig.12 and Fig.11. Our findings indicate that an increase in D_v significantly enhances the visibility and definition of Turing patterns for v , making them more obvious. On the contrary, the Turing patterns for u appear to be relatively invariant to changes in D_v .

In reaction-diffusion systems like Turing models, the diffusion coefficient D often plays a crucial role. A larger D_v allows the substance v to spread more rapidly in space, which in turn helps in forming clearer or more obvious patterns. A higher D_v could enhance the system's stability or increase the contrast of the pattern, making the Turing patterns for v easier to observe and identify. The Turing pattern for u is nearly invariant to D_v , it could imply that u primarily has a local role in the model, or that the diffusion and reaction rates for u are different enough from v that it is insensitive to changes in D_v .

6.5 Gierer-Meinhardt model for sphere

The Gierer-Meinhardt model has been widely studied for its role in describing complex pattern formation in biological systems and other phenomena. While a substantial body of research exists on Turing patterns in unbounded or flat domains, less work has involved applying the model to curved surfaces, such as spheres. When applied to spherical domains, the Gierer-Meinhardt model offers a unique approach not only for studying the conditions that lead to Turing pattern formation but also for examining how these patterns evolve over time on curved surfaces. The curvature of the sphere itself adds an additional layer of complexity to pattern behavior. This curvature could have a significant impact on the stability, shape, and complexity of Turing patterns and may offer many new phenomena for exploration.

As shown in Fig.13 and Fig.14. Here, we investigate the impact of σ_v . σ_v is a parameter directly related to the generation or decay of the variable v . When we increase the value of σ_v from 0.1 to 2, both u and v were observed to become unstable and reach larger numerical values.

This phenomenon can be interpreted as follows: A larger value of σ_v implies a faster rate of generation for v , which in turn amplifies the effect of the interaction term $\frac{u^2 v}{1+ku^2}$ between u and v . As this term becomes more pronounced, the system is more susceptible to perturbations, leading to the observed instability in u and v . Additionally, this instability may result in larger numerical values for both u and v . This observation provides deeper insights into the dynamic behavior of the model, especially in how the parameter σ_v influ-

Impact of D_v $\begin{cases} \frac{\partial u}{\partial t} = D_u \nabla^2 u + \rho_u \frac{u^2 v}{1+ku^2} - \mu_u u, \\ \frac{\partial v}{\partial t} = D_v \nabla^2 v - \rho_v \frac{u^2 v}{1+ku^2} + \sigma_v, \end{cases}$

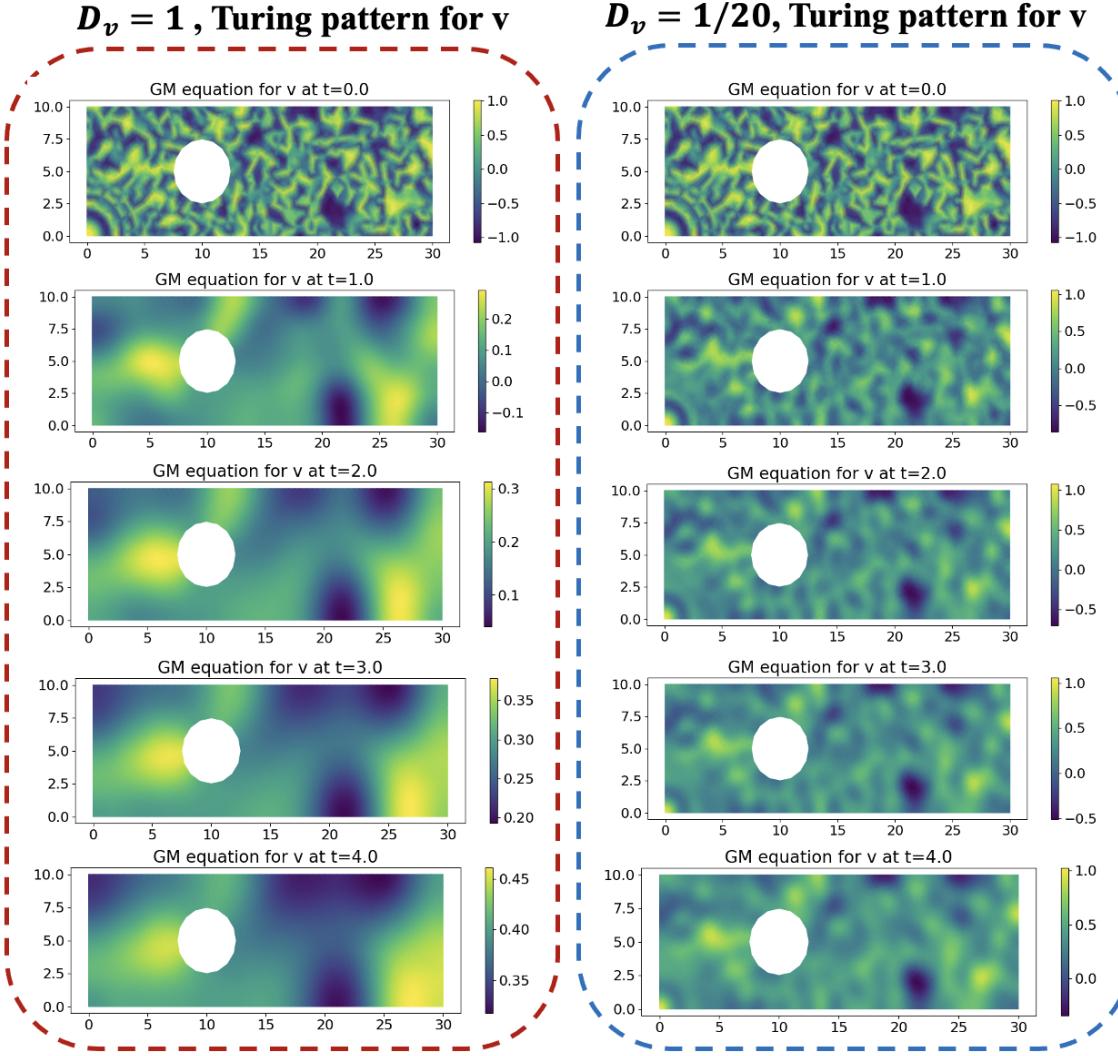


Figure 11: Gierer-Meinhardt Schnakenberg model for square with hole for v . We then calculate the equation for two different $D_v = 1$ or $D_v = 1/20$. The other parameters satisfy $D_u = 1/20$, $k = 0$, $\rho_u = 0.18$, $\rho_v = 0.36$, $u = 0.08$ and $\sigma_v = 0.1$. As shown in figure, we found that the larger the D_v , the turing pattern for v would be more obvious.

$$\text{Impact of } D_v \quad \left\{ \begin{array}{l} \frac{\partial u}{\partial t} = D_u \nabla^2 u + \rho_u \frac{u^2 v}{1+ku^2} - \mu_u u, \\ \frac{\partial v}{\partial t} = D_v \nabla^2 v - \rho_v \frac{u^2 v}{1+ku^2} + \sigma_v, \end{array} \right.$$

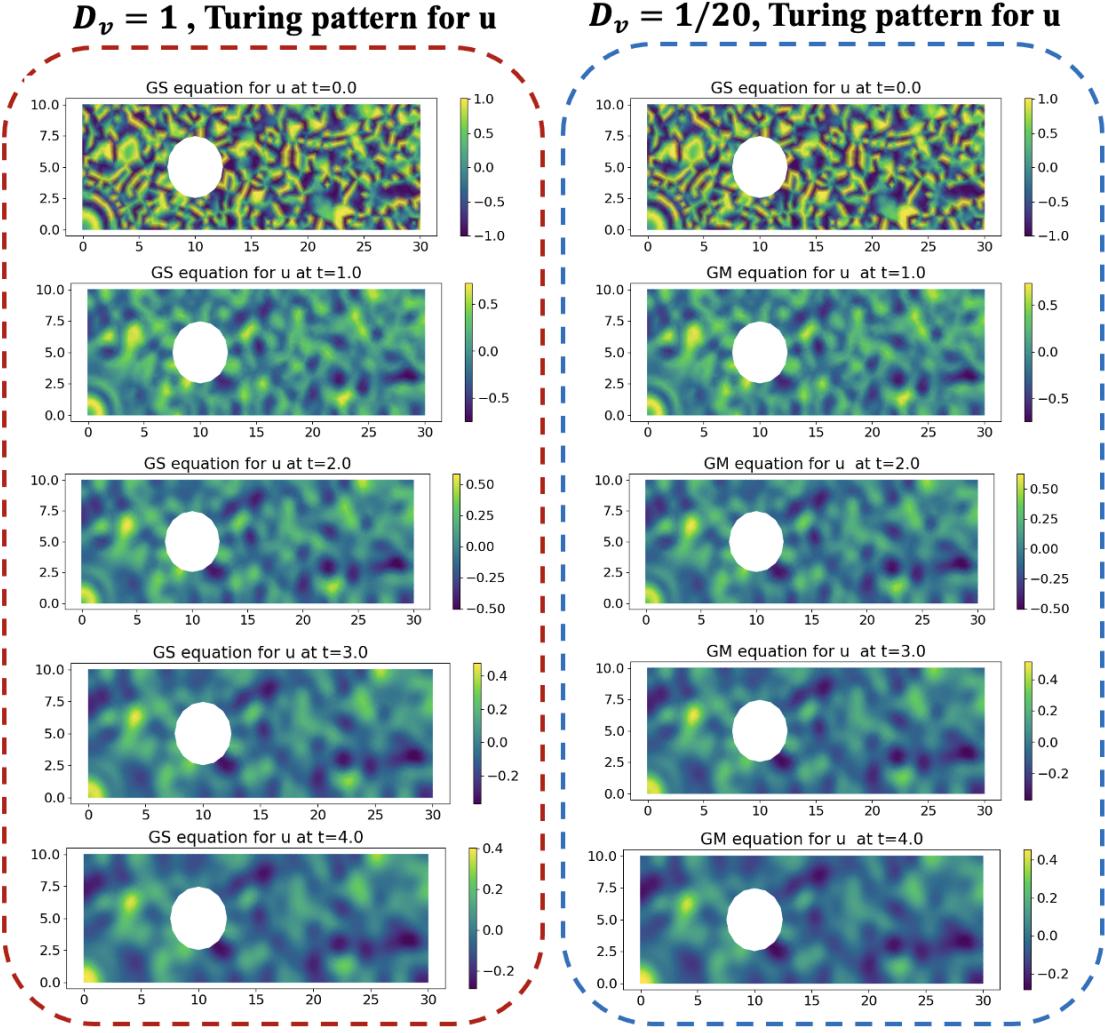


Figure 12: Gierer-Meinhardt Schnakenberg model for square with hole for u . We then calculate the equation for two different $D_v = 1$ or $D_v = 1/20$. The other parameters satisfy $D_u = 1/20$, $k = 0$, $\rho_u = 0.18$, $\rho_v = 0.36$, $u = 0.08$ and $\sigma_v = 0.1$. As shown in figure, the pattern of u is nearly irrelavent for D_v .

ences the stability and characteristics of the solution. This understanding could be crucial for controlling or optimizing systems described by this type of dynamical equations.

6.6 Gierer-Meinhardt model for torus

When applied to a torus, the Gierer-Meinhardt model also provides a deeply insightful research framework. Unlike flat or spherical surfaces, the topology and curvature of a torus introduce more complex variables into pattern formation. This complexity could lead to a range of new or different Turing patterns that are distinct from those on other geometric shapes.

We conducted further analysis on the Gierer-Meinhardt Schnakenberg model applied to torus, focusing particularly on the role of the parameter ρ_u in pattern formation. Two different values of ρ_u , namely 0.18 and 1.8, were examined while keeping other parameters consistent: $D_u = 1/20$, $D_v = 1$, $k = 0$, $\rho_v = 0.36$, $\mu_u = 0.08$, and $\sigma_v = 0.1$.

Our simulations reveal that the Turing pattern of u is more stable when ρ_u is larger. This can be attributed to the fact that a larger ρ_u essentially influences the rate of production or decay for u . In this context, a higher ρ_u leads to a more stable state for u , perhaps by balancing the nonlinear interaction terms more effectively, thereby stabilizing the pattern. Interestingly, the Turing pattern for v shows almost no sensitivity to changes in ρ_u . This insensitivity could be explained by the specific roles u and v play in the model equations. The rate of change of v might be more significantly controlled by other parameters such as ρ_v or σ_v , making it less susceptible to changes in ρ_u .

The reason is a higher ρ_u directly impacts the rate of change for u , making the system more resilient to perturbations and therefore more stable. The increased stability could also be a result of the balance between the generation and decay terms in the equation, which is influenced by ρ_u . For v , the rate of change for v might be more directly affected by parameters other than ρ_u , such as ρ_v or σ_v .

7 Numerical results of FitzHugh-Nagumo model

7.1 Introduction to FitzHugh-Nagumo model

The FitzHugh-Nagumo (FN) model [11] serves as a foundational concept in a wide range of scientific disciplines including mathematical biology, neuroscience, and physics. Initially conceived as a simplified alternative to the Hodgkin-Huxley equations, the model condenses the dynamics of excitable and oscillatory systems, such as neurons and cardiac

Impact of σ_v

$$\begin{cases} \frac{\partial u}{\partial t} = D_u \nabla^2 u + \rho_u \frac{u^2 v}{1+ku^2} - \mu_u u, \\ \frac{\partial v}{\partial t} = D_v \nabla^2 v - \rho_v \frac{u^2 v}{1+ku^2} + \sigma_v, \end{cases}$$

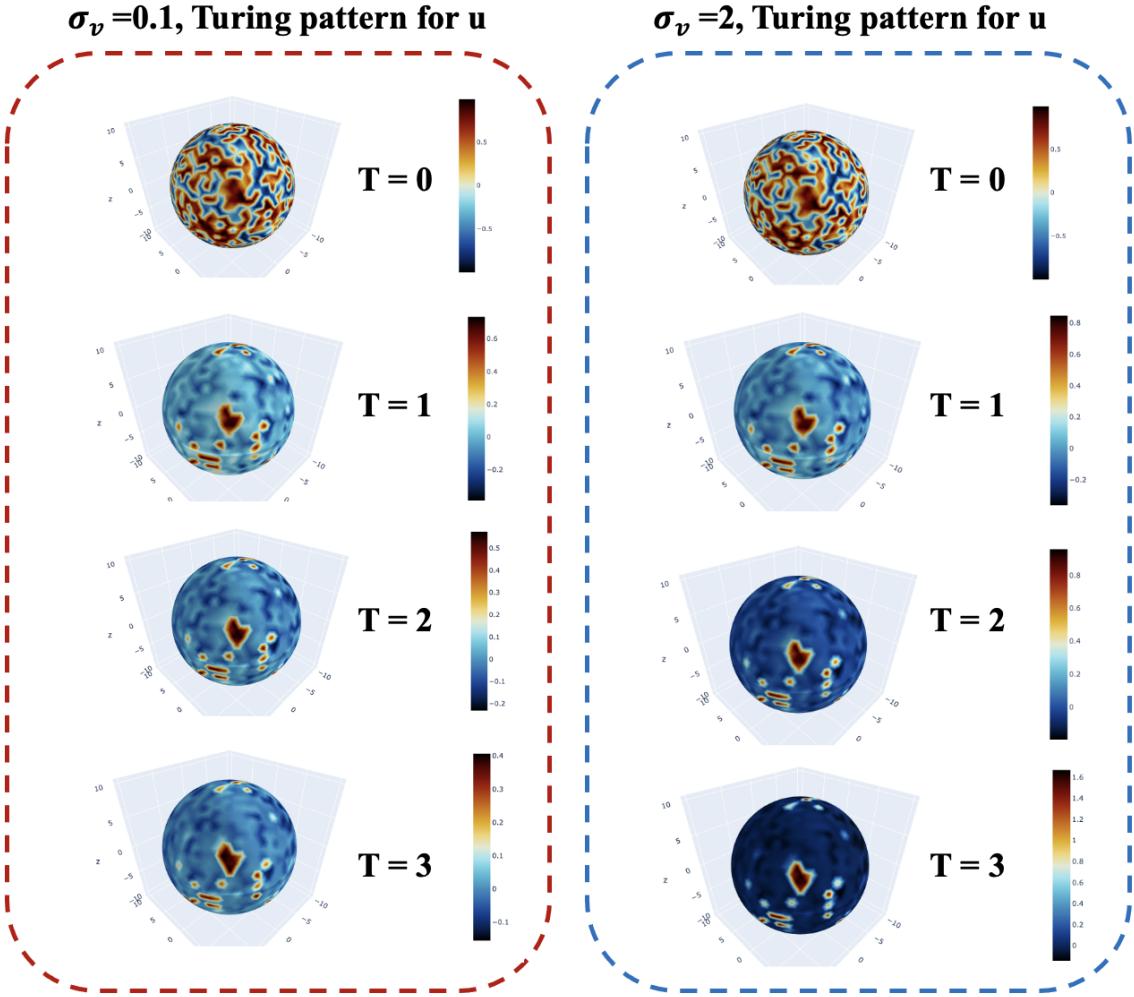


Figure 13: Gierer-Meinhardt Schnakenberg model for sphere surface for u . We then calculate the equation for two different $\sigma_v = 0.1$ or $\sigma_v = 2$. The other parameters satisfy $D_u = 1/20$, $D_v = 1$, $k = 0$, $\rho_u = 0.18$, $\rho_v = 0.36$ and $u = 0.08$. As shown in figure, the pattern of u would be affect due to the decrease of v .

Impact of σ_v

$$\begin{cases} \frac{\partial u}{\partial t} = D_u \nabla^2 u + \rho_u \frac{u^2 v}{1+ku^2} - \mu_u u, \\ \frac{\partial v}{\partial t} = D_v \nabla^2 v - \rho_v \frac{u^2 v}{1+ku^2} + \sigma_v, \end{cases}$$

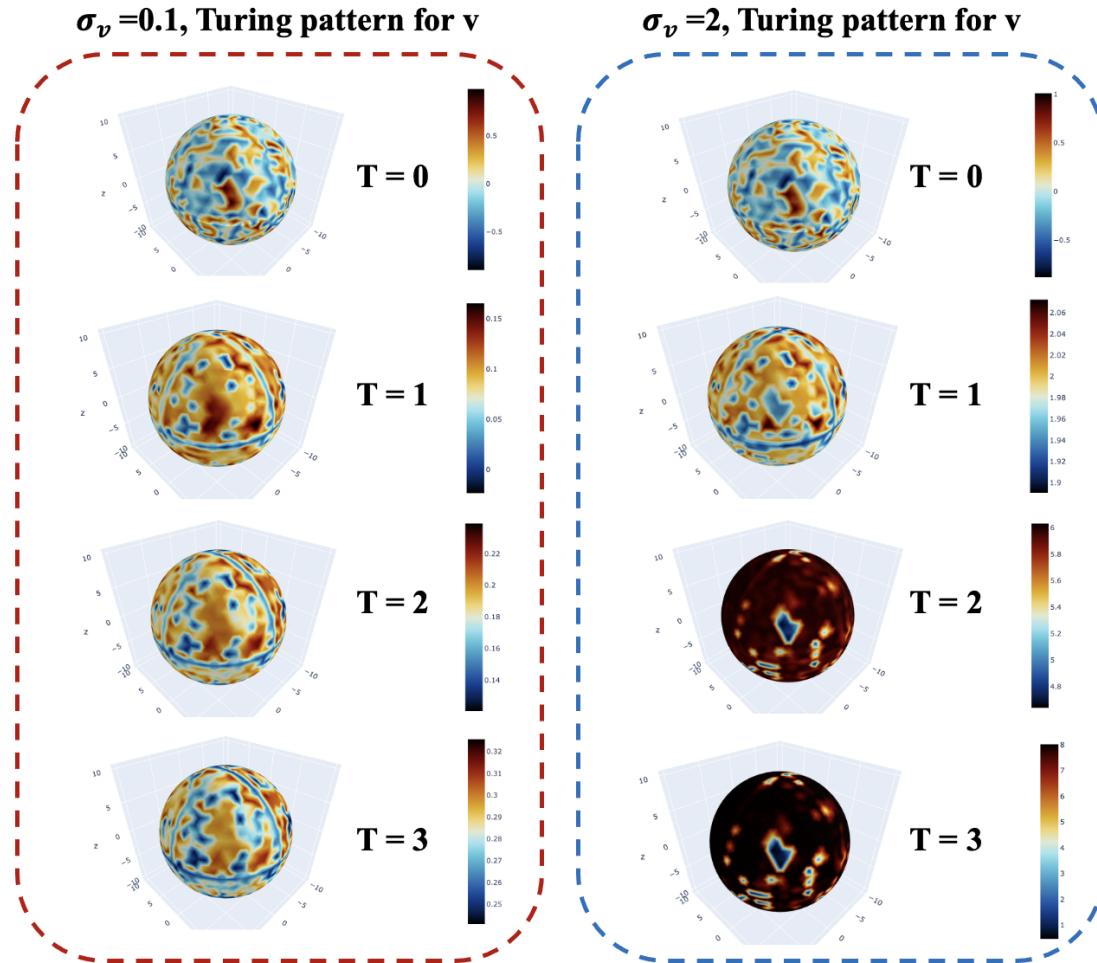


Figure 14: Gierer-Meinhardt Schnakenberg model for sphere surface for v . We then calculate the equation for two different $\sigma_v = 0.1$ or $\sigma_v = 2$. The other parameters satisfy $D_u = 1/20$, $D_u = 1$, $k = 0$, $\rho_u = 0.18$, $\rho_v = 0.36$ and $u = 0.08$. As shown in figure, the pattern of v would be decrease due to large value of σ_v .

Impact of ρ_u

$$\begin{cases} \frac{\partial u}{\partial t} = D_u \nabla^2 u + \rho_u \frac{u^2 v}{1+ku^2} - \mu_u u, \\ \frac{\partial v}{\partial t} = D_v \nabla^2 v - \rho_v \frac{u^2 v}{1+ku^2} + \sigma_v, \end{cases}$$

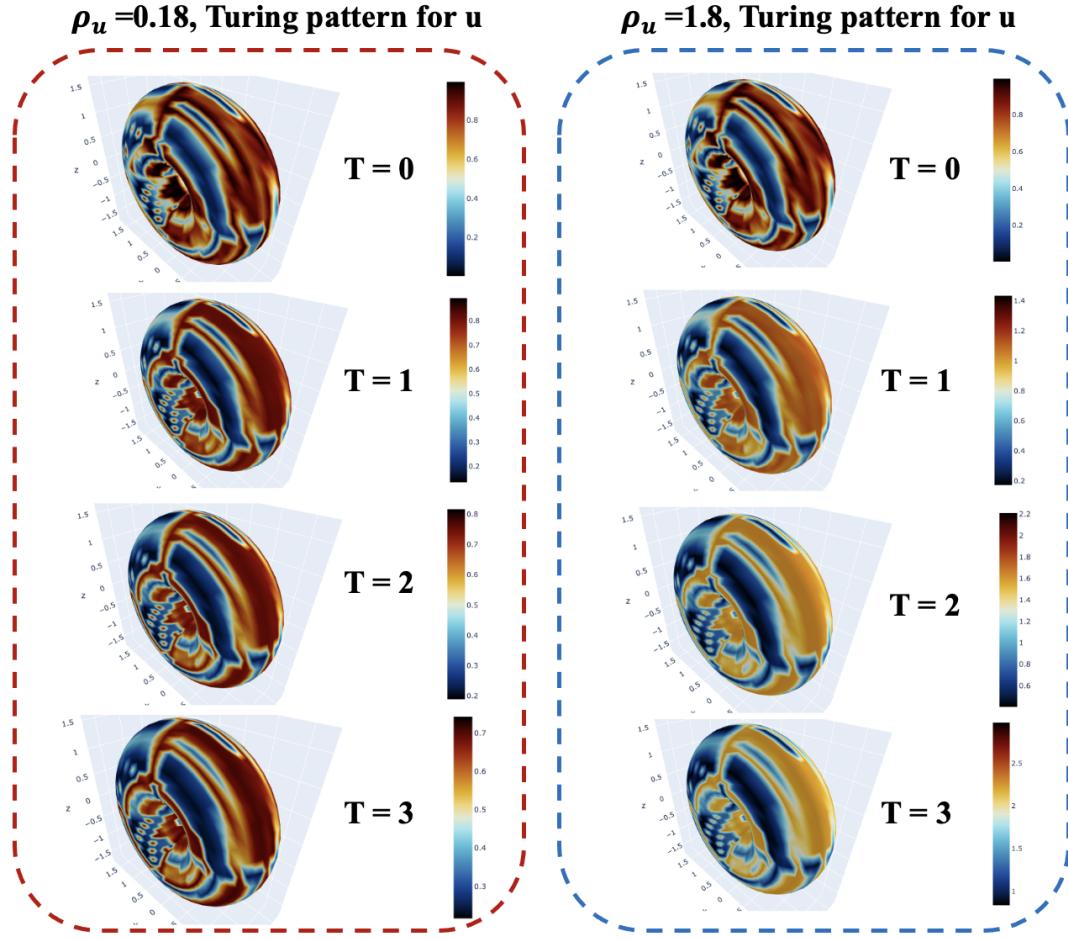


Figure 15: Gierer-Meinhardt Schnakenberg model for torus for u . We then calculate the equation for two different $\rho_u = 0.18$ or $\rho_u = 1.8$. The other parameters satisfy $D_u = 1/20, D_u = 1, k = 0, \rho_v = 0.36, u = 0.08$ and $\sigma_v = 0.1$. As shown in figure, the pattern of u is more stable with larger ρ_u .

Impact of ρ_u

$$\begin{cases} \frac{\partial u}{\partial t} = D_u \nabla^2 u + \rho_u \frac{u^2 v}{1+ku^2} - \mu_u u, \\ \frac{\partial v}{\partial t} = D_v \nabla^2 v - \rho_v \frac{u^2 v}{1+ku^2} + \sigma_v, \end{cases}$$

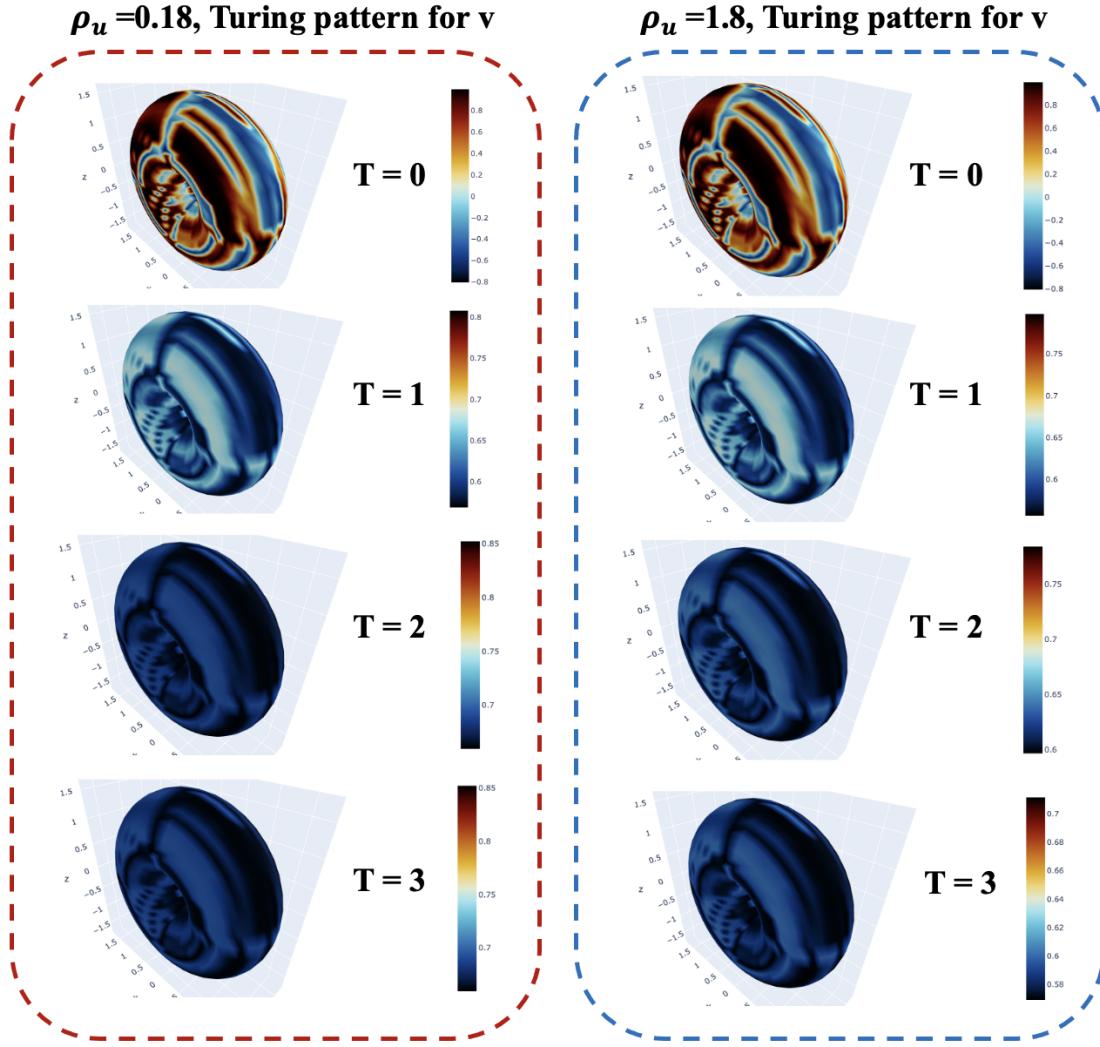


Figure 16: Gierer-Meinhardt Schnakenberg model for torus for u . We then calculate the equation for two different $\rho_u = 0.18$ or $\rho_u = 1.8$. The other parameters satisfy $D_u = 1/20$, $D_u = 1$, $k = 0$, $\rho_v = 0.36$, $\mu_u = 0.08$ and $\sigma_v = 0.1$. As shown in figure, the pattern of v is nearly irrelavent for ρ_u .

cells, into two coupled non-linear differential equations. Given its computational efficiency and versatility, the model has permeated diverse research areas like neuroscience, cardiology, ecology, and chemical kinetics. In this section, we delve into the impact of parameter variations on the final solutions of the FitzHugh-Nagumo model, enriching our understanding of the system's dynamics.

The FitzHugh-Nagumo model has been widely employed to study diverse phenomena such as the firing patterns of neurons, cardiac arrhythmias, and pattern formation in chemical reactions. The model's mathematical simplicity makes it accessible for analytical treatment, while its capacity to encapsulate essential dynamics makes it a robust tool for computational simulations. In spatially extended systems, it has also been used to describe wavefront dynamics and pattern formation, features that are integral to many biological and physical phenomena.

Understanding how different parameters affect the final solutions in the FN model is crucial for its applications in various fields [12]. In medical contexts, such as neurobiology and cardiology, fine-tuning these parameters can offer insights into disease mechanisms or potential therapeutic interventions. Likewise, in ecological models, understanding parameter sensitivity can help predict population dynamics under various environmental conditions.

7.2 Mathematical equation of FitzHugh-Nagumo Model

In a spatially extended context, the FitzHugh-Nagumo model can be expressed as follows:

$$\begin{cases} \frac{\partial u}{\partial t} = D_u \nabla^2 u + \left(u - \frac{u^3}{3} + v - 0.6 \right), \\ \frac{\partial v}{\partial t} = D_v \nabla^2 v - (u - 0.6 + 9.99v), \end{cases}$$

Here, u and v are the "activator" and "inhibitor" variables, respectively, and their temporal and spatial evolution are dictated by the equations. The term $\frac{\partial}{\partial t}$ describes the change over time, while ∇^2 represents the Laplacian operator, accounting for spatial diffusion.

The coefficients D_u and D_v signify the diffusion rates for u and v , respectively, influencing how these variables spread over space. The nonlinear terms $u - \frac{u^3}{3} + v - 0.6$ and $u - 0.6 + 9.99v$ model the intrinsic dynamics of the activator and inhibitor variables. These terms capture the inherent complexities of the system, such as the feedback mechanisms and threshold-like behaviors that are commonly observed in biological systems.

7.3 Weak form of FitzHugh-Nagumo model

The FitzHugh-Nagumo (FN) model into its weak form, similarly to how the Gierer-Meinhardt model was treated. The FN model equations in their strong form are:

$$\begin{cases} \frac{\partial u}{\partial t} = D_u \nabla^2 u + \left(u - \frac{u^3}{3} + v - 0.6 \right), \\ \frac{\partial v}{\partial t} = D_v \nabla^2 v - (u - 0.6 + 9.99v), \end{cases}$$

To proceed to the weak form using Firedrake and UFL, we first multiply each equation by corresponding test functions ϕ for u and ψ for v , and integrate over the domain Ω .

For the u equation:

$$\int_{\Omega} \phi \frac{\partial u}{\partial t} dx = \int_{\Omega} \phi D_u \nabla^2 u dx + \int_{\Omega} \phi \left(u - \frac{u^3}{3} + v - 0.6 \right) dx$$

Applying integration by parts to the Laplacian term gives:

$$\int_{\Omega} \phi D_u \nabla^2 u dx = - \int_{\Omega} D_u \nabla \phi \cdot \nabla u dx$$

Similarly, for the v equation:

$$\int_{\Omega} \psi \frac{\partial v}{\partial t} dx = \int_{\Omega} \psi D_v \nabla^2 v dx + \int_{\Omega} \psi (-u + 0.6 - 9.99v) dx$$

After applying integration by parts:

$$\int_{\Omega} \psi D_v \nabla^2 v dx = - \int_{\Omega} D_v \nabla \psi \cdot \nabla v dx$$

Now, combining all these, we get the weak formulations of the equations as:

$$\begin{cases} \int_{\Omega} \phi \frac{\partial u}{\partial t} dx + \int_{\Omega} D_u \nabla \phi \cdot \nabla u dx - \int_{\Omega} \phi \left(u - \frac{u^3}{3} + v - 0.6 \right) dx = 0, \\ \int_{\Omega} \psi \frac{\partial v}{\partial t} dx + \int_{\Omega} D_v \nabla \psi \cdot \nabla v dx - \int_{\Omega} \psi (-u + 0.6 - 9.99v) dx = 0. \end{cases}$$

These weak forms are suitable for numerical approximation using finite element methods. You can now proceed with numerical simulations of the FN model using Firedrake or other computational tools that employ the finite element method. And the python version can be represented as:

```

1 F_u = (u - u0)*phi*dx - dt*(-D_u*dot(grad(u), grad(phi))*dx
   + u*phi*dx - u*u*u/3*phi*dx+v*phi*dx-0.6*phi*dx)
2 F_v = (v - v0)*psi*dx - dt*(-D_v*dot(grad(v), grad(psi))*dx
   - u*psi*dx + 0.6*psi*dx-9.99*psi*dx)
```

7.4 FitzHugh-Nagumo model for square with hole

As shown in Fig.17 and Fig.18. After conducting tests on the FitzHugh-Nagumo model's final solutions for u and v with varying diffusion coefficients D_u and D_v , specifically setting $D_u = 1, D_v = 1.75$ and $D_u = 0.0023, D_v = 0.0045$, we observed distinct differences in the final solution patterns. Higher values of D_u and D_v resulted in more stable final solution patterns, while lower values of D_u and D_v preserved greater complexity in the patterns of the final solutions.

The magnitude of D_u and D_v will dictate how rapidly u and v spread out in space. Larger diffusion coefficients imply faster diffusion, resulting in more "smoothed out" or stable patterns as the system evolves. Smaller coefficients mean that u and v will tend to stay more localized, preserving complex patterns that might form.

The diffusion term essentially serves as a "smoothing" operator over time. Higher diffusion rates lead to quicker equalization of concentration levels of u and v across the domain, making the system more stable. Lower diffusion rates are less effective at smoothing out these variations, which allows for the preservation of more complex and nuanced patterns.

7.5 FitzHugh-Nagumo model for sphere

The FitzHugh-Nagumo (abbreviated as FN) model was originally developed to describe neuronal activity but has found applications in multiple fields such as biology, chemistry, and physics. When applied to a sphere—a three-dimensional curved surface—the model provides an interesting and challenging framework for understanding the formation and evolution of complex patterns.

On a three-dimensional sphere, the behavior of the model is subject to additional geometric constraints and the curvature of the sphere itself. This can lead to unique phenomena, such as different stabilities, shapes, and complexities of the patterns. Curvature and surface tension could have pronounced effects on spatial patterns, effects that would not be present in flat or unbounded domains.

As shown in Fig.19, when $D_u = 1$ and $D_v = 1.75$, the high diffusion rates serve to stabilize the pattern by allowing u and v to disperse quickly. This rapid dispersion minimizes local variations, making the pattern more uniform and stable over time. And when $D_u = 0.0023$ and $D_v = 0.0045$, the diffusion is much slower, thereby preserving local variations and complexities in the pattern. The system becomes more susceptible to initial conditions and small perturbations, leading to more intricate and complex patterns in the final solution.

Final solution for different parameters

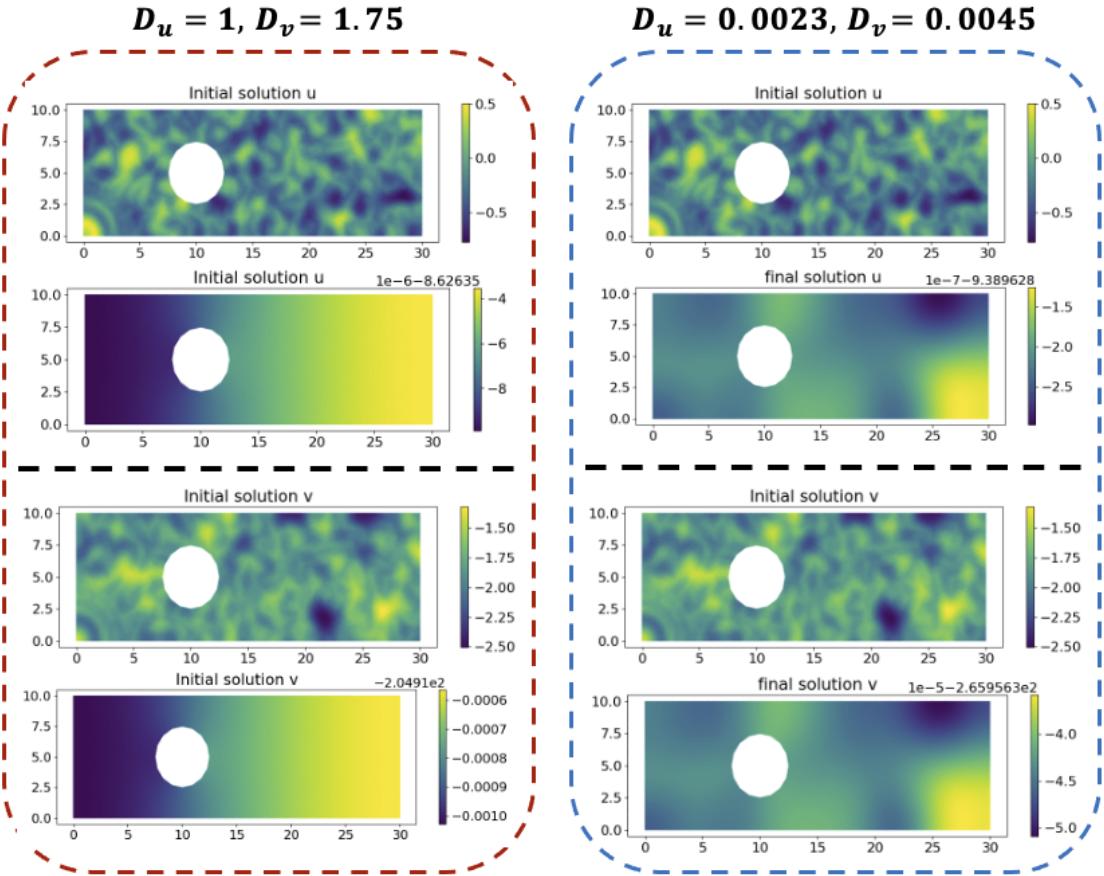


Figure 17: FitzHugh-Nagumo model final solution for u and v on square with hole. After Testing the $D_u = 1$, $D_v = 1.75$ and $D_u = 0.0023$ and $D_v = 0.0045$. We found that the final solution pattern is different. The larger value of D_u and D_v would cause the final solution to become more stable and the smaller value of D_u and D_v would save more pattern on final solution.

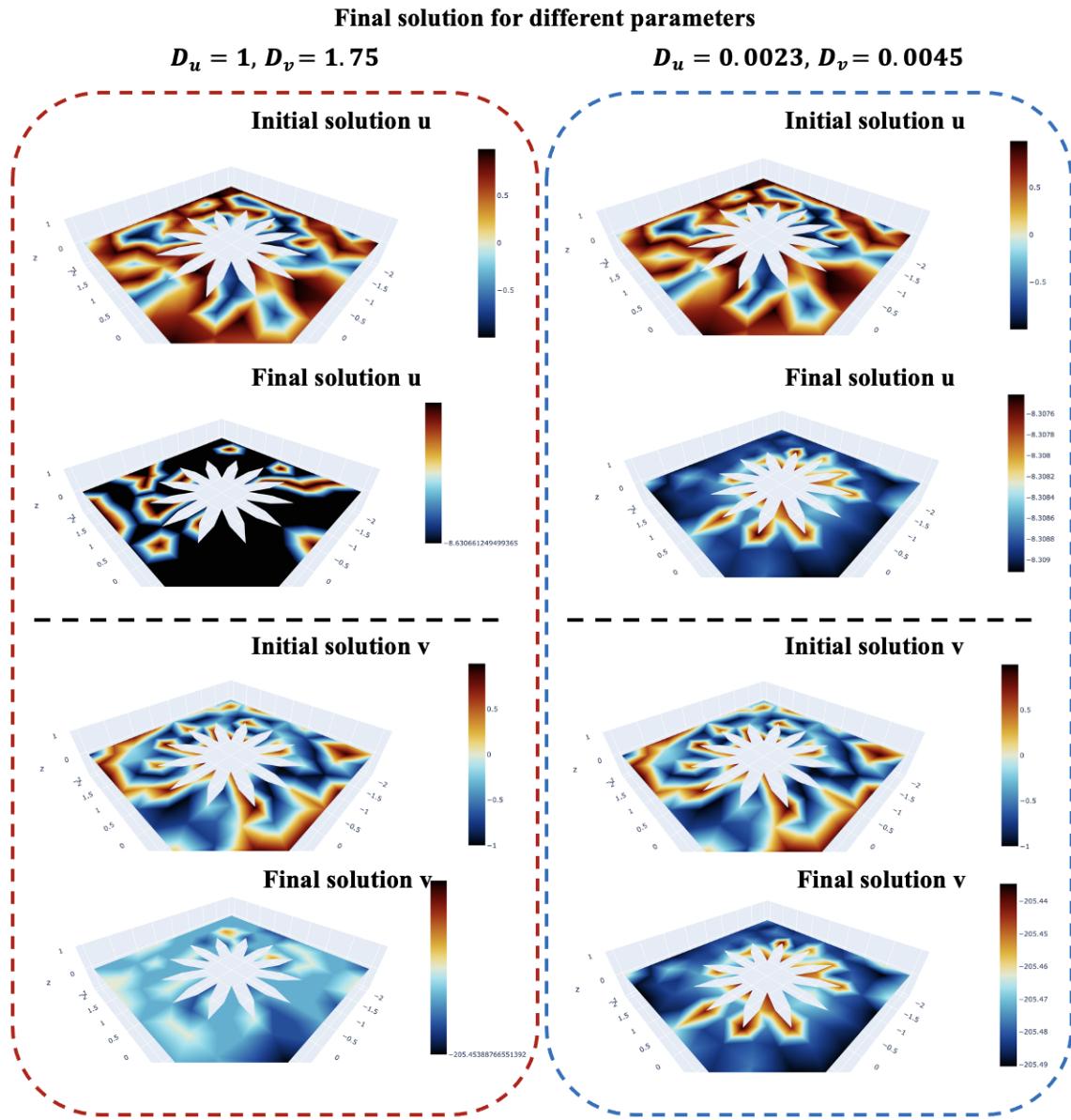


Figure 18: FitzHugh-Nagumo model final solution for u and v on square with flower. After Testing the $D_u = 1, D_v = 1.75$ and $D_u = 0.0023$ and $D_v = 0.0045$. We found that the final solution pattern is different. The larger value of D_u and D_v would cause the final solution to become more stable and the smaller value of D_u and D_v would save more pattern on final solution.

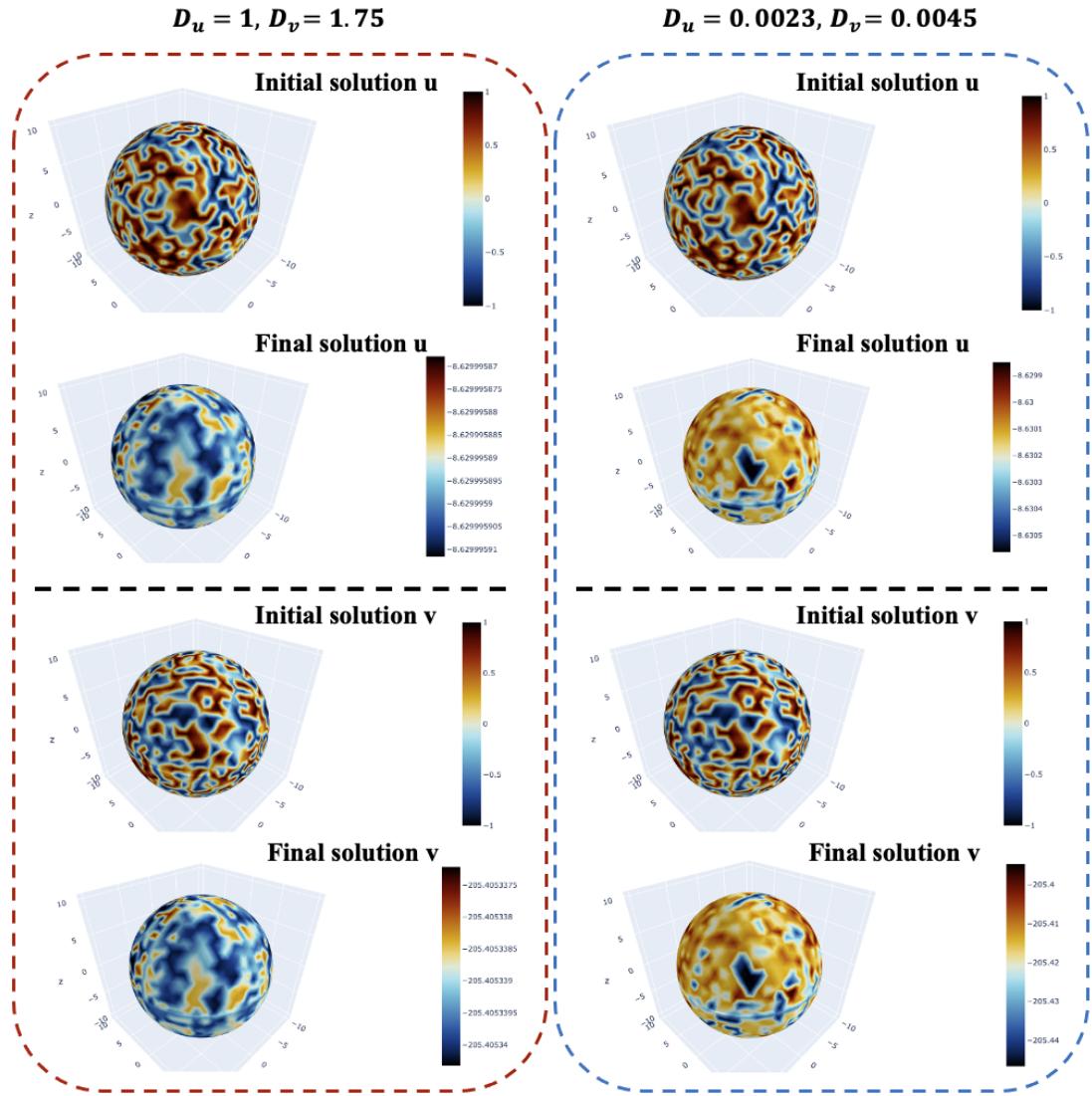


Figure 19: FitzHugh-Nagumo model final solution for u and v on sphere. After Testing the $D_u = 1$, $D_v = 1.75$ and $D_u = 0.0023$ and $D_v = 0.0045$. We found that the final solution pattern is different. The larger value of D_u and D_v would cause the final solution to become more stable and the smaller value of D_u and D_v would save more pattern on final solution.

7.6 FitzHugh-Nagumo model for torus

The FitzHugh-Nagumo model has wide applications in multiple scientific domains, including biology, physics, and chemistry. When this model is applied to a torus surface , it opens up a new and highly challenging problem.

As shown in Fig.20, when $D_u = 1$ and $D_v = 1.75$, the high diffusion rates serve to stabilize the pattern by allowing u and v to disperse quickly. This rapid dispersion minimizes local variations, making the pattern more uniform and stable over time. And when $D_u = 0.0023$ and $D_v = 0.0045$, the diffusion is much slower, thereby preserving local variations and complexities in the pattern. The system becomes more susceptible to initial conditions and small perturbations, leading to more intricate and complex patterns in the final solution.

The terms $D_u \nabla^2 u$ and $D_v \nabla^2 v$ represent the diffusion of u and v , respectively. The magnitude of D_u and D_v will dictate how rapidly u and v spread out in space. Larger diffusion coefficients imply faster diffusion, resulting in more "smoothed out" or stable patterns as the system evolves. Smaller coefficients mean that u and v will tend to stay more localized, preserving complex patterns that might form.

In summary, the diffusion coefficients D_u and D_v play a critical role in determining the stability and complexity of the patterns in the FitzHugh-Nagumo model.

8 Compare three models

The Schnakenberg model, Gierer-Meinhardt model, and FitzHugh-Nagumo (FHN) model are mathematical models that are commonly used in various scientific disciplines, including biology, chemistry, and physics, to describe pattern formation, chemical reactions, and excitable systems.

8.1 Similarities

All of these models can generate patterns under certain conditions, such as Turing patterns, making them applicable for studying a wide variety of pattern formation phenomena. These equations describe how two or more variables, usually concentrations or activation levels, change over time and space.

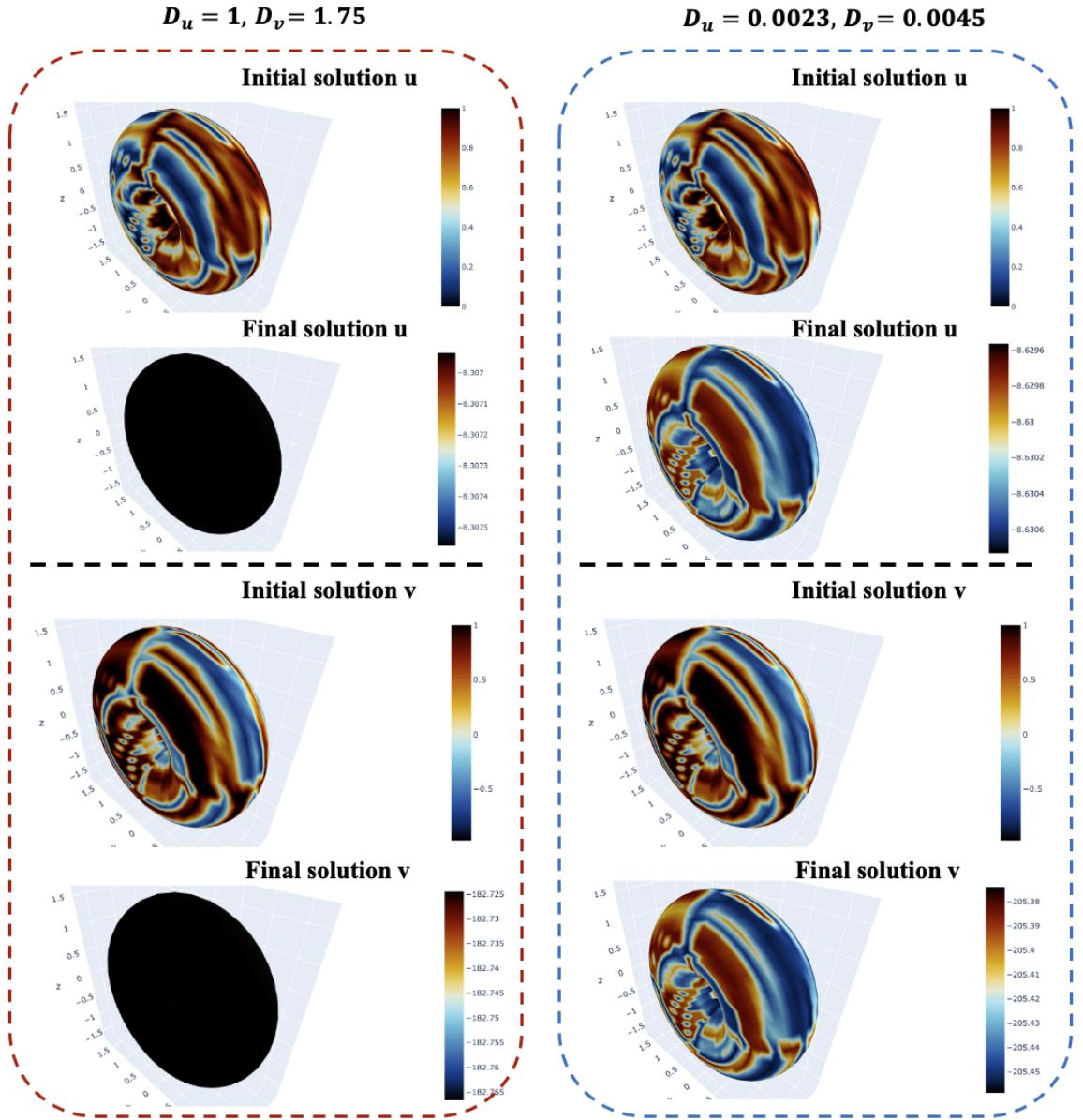


Figure 20: FitzHugh-Nagumo model final solution for u and v on torus. After Testing the $D_u = 1, D_v = 1.75$ and $D_u = 0.0023$ and $D_v = 0.0045$. We found that the final solution pattern is different. The larger value of D_u and D_v would cause the final solution to become more stable and the smaller value of D_u and D_v would save more pattern on final solution.

8.2 Differences

First, the Schnakenberg model is primarily applied in chemistry to describe autocatalytic reactions and has also found applications in biological pattern formation. The Gierer-Meinhardt model is mainly used in developmental biology, such as for explaining animal fur patterns. The FitzHugh-Nagumo model is most commonly employed in neuroscience and cardiology, particularly to describe excitable and oscillatory systems like neurons and cardiac cells.

Second, the Schnakenberg model is relatively less sensitive to parameter changes, making it more robust for certain applications. The Gierer-Meinhardt model is more sensitive due to its complex nonlinearities. The sensitivity in the FitzHugh-Nagumo model can vary based on the specific form of parameters. In our visualization process, we can also find that the Schnakenberg model is relatively less sensitive, Gierer-Meinhardt model is more sensitive and FitzHugh-Nagumo model can vary based on the specific form of parameters.

9 Conclusion

In this thesis, we have conducted a comprehensive exploration into the application of the finite element method for solving the Schnakenberg equation, Gierer-Meinhardt model, and FitzHugh-Nagumo model. The paper consists of nine chapters, spanning from the use of software tools to the numerical solving and parameter analysis of complex models. Chapter 1 introduces the fundamental knowledge of turing pattern. Chapter 2 introduces the three major software tools involved in implementing the full finite element scheme, namely Docker, Gmsh, and Firedrake. Chapter 3 elaborates on how to use Gmsh to create various surfaces, including planes with holes, spheres, cylinders, and tori. Chapter 4 primarily discusses the basic installation and usage of Firedrake, demonstrating the transformation of the Schnakenberg equation into its weak form and solving it using the finite element method. Chapter 5 investigates the visualization results of the numerical solutions for the Schnakenberg equation on surfaces with holes, spheres, and tori. Chapter 6 takes a closer look at how parameters in the Gierer-Meinhardt model influence the numerical solutions on these same geometrical surfaces. Chapter 7 further explores the relationship between the steady-state solutions and parameters in the FitzHugh-Nagumo model on these surfaces. Chapter 8 compares the different models, highlighting the similarities and differences between them. Finally, in Chapter 9, we summarize the key findings of the entire paper and propose directions for future research.

Not only does this thesis offer a comprehensive method for investigating the numerical

solutions of complex models, but it also demonstrates the flexibility and potential of these models in practical applications through their application on various geometric surfaces. This lays a solid foundation for future research.

9.1 Main contributions of the research

Our main contributions are as follows:

- Demonstrated proficiency in utilizing cutting-edge software tools such as Gmsh, Docker, and Firedrake. Leveraging these tools, we successfully implemented the finite element method to solve equations describing Turing patterns on curved surfaces in \mathbb{R}^3 . Our solutions yield numerical results that conform to the principles of Turing patterns, expanding the applicability of these models.
- Delve into the impact of various parameters on the numerical solutions of three different models at each time point on the curved surfaces. Our exploration has deepened the understanding of the role of these parameters within the Schnakenberg equation, Gierer-Meinhardt model, and FitzHugh-Nagumo model, thereby enhancing the predictive power and interpretability of these models.
- Provide insightful analysis on how diffusion coefficients and other parameters affect the final steady-state solutions on curved surfaces. This aspect of our work is crucial for gaining a more nuanced understanding of the systems being modeled and is instrumental in guiding future experimental designs and theoretical studies.

Overall, our work serves as a comprehensive methodological guide for those interested in Turing patterns, parameter analysis, and finite element methods in curved geometries. We have laid a strong foundation for future research in this rapidly evolving field.

9.2 Future directions and recommendations for research

Further, since this thesis focuses on Turing patterns on curved surfaces in \mathbb{R}^3 , future work could extend these methods to manifolds in higher-dimensional spaces. Understanding how Turing patterns evolve in these more complex environments could have interesting implications.

Besides, the parameter study would be to employ optimization algorithms to determine the 'best' set of parameters for a given application. This could have immediate practical applications in fields as diverse as drug delivery and material science.

References

- [1] Turing, A. M. The chemical basis of morphogenesis. *Bulletin of mathematical biology* **52**, 153–197 (1990).
- [2] Dufiet, V. & Boissonade, J. Conventional and unconventional turing patterns. *The Journal of chemical physics* **96**, 664–673 (1992).
- [3] Plaza, R. G., Sanchez-Garduno, F., Padilla, P., Barrio, R. A. & Maini, P. K. The effect of growth and curvature on pattern formation. *Journal of dynamics and differential equations* **16**, 1093–1121 (2004).
- [4] Nampoothiri, S. Stability of patterns on thin curved surfaces. *Physical Review E* **94**, 022403 (2016).
- [5] Murray, J. D. *Mathematical biology II: Spatial models and biomedical applications*, vol. 3 (Springer New York, 2001).
- [6] Docker, I. Docker. *linea].[Junio de 2017]. Disponible en: <https://www.docker.com/what-docker>* (2020).
- [7] Geuzaine, C. & Remacle, J.-F. Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering* **79**, 1309–1331 (2009).
- [8] Rathgeber, F. *et al.* Firedrake: automating the finite element method by composing abstractions. *ACM Transactions on Mathematical Software (TOMS)* **43**, 1–27 (2016).
- [9] Wallwork, J. G., Barral, N., Ham, D. A. & Piggott, M. D. Anisotropic goal-oriented mesh adaptation in firedrake. *28th Intl Meshing Roundtable, Zenodo* 83–100 (2020).
- [10] Gierer, A. & Meinhardt, H. A theory of biological pattern formation. *Kybernetik* **12**, 30–39 (1972).
- [11] Sánchez-Garduno, F., Krause, A. L., Castillo, J. A. & Padilla, P. Turing–hopf patterns on growing domains: the torus and the sphere. *Journal of theoretical biology* **481**, 136–150 (2019).
- [12] Glowinski, R. Finite element methods for incompressible viscous flow. *Handbook of numerical analysis* **9**, 3–1176 (2003).