國立交通大學

應用數學系

碩士論文

反應擴散方程在移動曲面上的數值模擬

Numerical Simulations of the Reaction-Diffusion Equation on the Moving Surface

研 究 生：陳建明

指導教授：賴明治　教授

中華民國九十七年七月

反應擴散方程在移動曲面上的數值模擬

Numerical Simulations of the Reaction-Diffusion Equation on the Moving Surface

研究生：陳建明　　Student: Chien-Ming Chen

指導教授：賴明治　　Advisor: Ming-Chih Lai

國立交通大學
應用數學系
碩士論文

A Thesis
Submitted to Department of Applied Mathematics
College of Science
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Master
in
Applied Mathematics
July 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年七月

反應擴散方程在移動曲面上的數值模擬

學生 ： 陳建明　　　　　　　　　　　　　　指導教授 ： 賴明治 教授

國立交通大學應用數學系(研究所)碩士班

摘　　要

這篇論文的主要目的是使用快速傅立葉轉換的優點來數值上地解反應擴散方程組，並且解以移動的曲面為定義域的方程組。在未來的工作上希望能夠找出近似心臟的舒張與收縮的移動曲面方程式以結合反應擴散方程式來做出更好的心臟電流傳導的數值模擬。首先，在球面座標與橢球面座標下利用譜方法和一些二階精確的數值方法分別做球面上及橢球面上的熱方程快速解法。接著，使用這兩種快速解法並且結合時間分解的方式產生反應擴散方程的快速解法。再來，使用在曲線座標下的曲面拉普拉斯算子來數值上地解熱方程，利用這個數值解可以計算定義在移動曲面上的熱方程，進一步用它來計算傳導擴散方程。在以上的各種數值解下，我們皆舉了四個可以涵蓋其他各種情形的例子去觀察質量的變化。最後，對這些數值解和質量的變化我們做出了結論及它們的應用性。

Numerical Simulations of the Reaction-Diffusion Equation on the Moving Surface

Student：Chien-Ming Chen　　Advisor：Dr.Ming-Chih Lai

Department(Institute) of Applied Mathematics
National Chiao Tung University

ABSTRACT

The central objective of this thesis is to use the fast fourier transform to numerically solve the reaction-diffusion systems and solve another systems whose domain moves with time. Hope that we could produce the equations on the moving domain which approximates the diastole and systole in human hearts. First, use the spectral method and some second-order numerical methods to produce the fast heat solvers on the spherical and ellipsoid surface domain in spherical coordinates and ellipsoid coordinates, respectively. Then, couple these two fast solvers with the time splitting method to produce the fast solver of the reaction-diffusion equation. Next, we use the surface Laplacian operator in Curvilinear coordinates to numerically compute the heat equation. Therefore, this heat solver could compute the heat equation on the moving surface domain. Furthermore, use it to compute the convection-diffusion equation. In each of above numerical solvers, we give four examples which could cover other situations to observe the changes in the mass. Finally, we summarize the applications and results for these numerical solvers and changes in the mass.

誌謝

這篇論文的完成，首先，我最需要感謝的是我的恩師，也就是我的指導教授賴明治教授。在老師的循循善誘下，我漸漸懂得數學在應用方面的實用性及重要性，對自己在學習的路上無疑是添加了強心劑，在作研究的過程中，不僅學會了許多數值計算的方法與實作，在科學計算上又有更深的了解；在每週的meeting裡，我學到的不僅僅是做研究的方法，還有做研究的態度，更重要的是做人處事的道理，在研究上他更是會替我們指引方向，在研究中的這片汪洋大海裡，老師有如指南針般讓我們不會在裡頭迷失自己要走的路，老師對我的重要性更是不言而喻。

在研究所這兩年裡，我要感謝曾昱豪與陳冠羽學長和曾孝捷學長，在我對研究與程式上遇到挫折、感到灰心、受到打擊時，有學長的聊天交流與學業和研究上的指點，其中得到的關懷讓我又對研究產生了許多衝勁，學長在我對研究的路上也指導了我許多正確的方向也給了我一些他的經驗法則，讓我在瓶頸中得到了許多啟發。

我要感謝308實驗室的所有室友(同學)，在平常的聚會或聊天中，有時都會聊到彼此研究的東西，互相以不同的角度來思考，說出自己的想法，這樣的實驗室讓我學習到team的重要，以及由team感受到大家用心在研究上的氣氛，對我這兩年的研究生活也添加了許多動力和想法，其中尤其是我的室友謝先皓，在我剛踏入交大應數所的研究之旅時，他以在交大應數系時的豐富經驗與數學知識給予我在研究與課業上許多的幫助，是個非常好的研究搭檔。

我要感謝吳恭儉學長,在我PDE方面學識較不足的地方以及對課業產生困惑的地方,總是有他可以讓我請教，並從中獲取新的idea。

我要感謝蔡明誠、李金龍與李信儀學長，在一年級的必修實變數函數論中，學長們會不吝惜自己的時間願意讓我與他討論課業上的問題，面對我的問題，他們總是不厭其煩的指導我，不管犧牲了多少學長自己的時間，他們都是一樣熱心，感謝學長的熱誠與用心，讓我能在有著必修壓力的碩一生活能順利度過，也感受到研究生活中令人感到相當溫暖的一面。

研究所的路上，還有著許多的學長姐與學弟妹溫暖的關心，像芳婷、蕙蘭、康伶學姐與喻培、明耀、文貴、育豪、川和學長與仁洲、裕昇、振庭學弟和實驗室的學弟們都是我能快樂享受研究的來源之一，讓我感受到作研究的我並不孤單。

再來，我要特別感謝的是這次擔任口試的委員王偉仲教授、黃聰明教授、吳金典教授，謝謝他們花費心思檢視我的論文，並且給予我的論文許多有益的建議，讓我的論文更趨完整。

最後，我感謝我的父母能夠讓我在這樣的研究所生涯裡，全力的去唸書與研究，不用去顧及其他事物的煩惱，以致於我能夠順利的畢業，沒有他們和女友怡玲的關心與期待，我將無法擁有這樣的動力與使命感去完成我的學業。也感謝所有關心我的人，在這裡與您一起分享此篇論文的喜悅。

目　　　錄

1、Introduction

Modeling is an imperative headway in science and numerical simulations stringing along with it. Both of them play the important roles in our modern livelihood. The reason is that the numerical simulations always help people to discover something amazing. For examples, they could be used to predict, test or find some rules, etc. In many applications of them, we concentrate on predicting the motions of waves. Propagation of waves has been appeared in many behaviors of biological and chemical experiments. Mathematical modeling can uphold and connect with these experiences. Because the electrical activities of the human hearts can be modeled by the nonlinear reaction-diffusion (RD) equations [15] and the FitzHugh -Nagumo-type model on the spherical surface domain with fixed radius has been simulated in [6], we are attracted to these related problems. In recent years, there are many kinds of cardiovascular diseases which our world replete with. We are anxious to develop some numerical simulations that are concerned with the electrical activities of our hearts to help treating the cardiopathy and anticipating obtaining some things good for the medical treatment of human beings. But these are our final expectations and goals. According to the developments of the present techniques of modeling, we only expect to use the forthcoming model in [6] to develop a mathematical basis for understanding the propagation of electrochemical waves in human hearts.

Here make mention of [6] in brief. The authors administered the simulation of the reaction-diffusion systems on a spherical surface domain with a radius $r \in [11,16]$ to perform circular waves by beginning with some initial values and regulating some parameters of the systems. These numerical results came into being meandering of spiral waves. Next, we are going to simulate the reaction-diffusion equation on a spherical surface domain with a radius $r = 16$ as [6] and simulate it on an ellipsoid surface domain whose size is close to a spherical surface with $r = 16$ and simulate it on an ellipsoid surface domain whose size is close to realistic cardiac in Chapter 5. Before performing our works, the numerical schemes in Chapter 2 are essential to our simulations. In Chapter 3, we solve the heat equation implicitly on a spherical surface domain in spherical coordinates with the schemes in Chapter 2 for Chapter 5. In Chapter 4, we solve the heat equation implicitly on an ellipsoid surface domain in ellipsoid coordinates with the schemes in Chapter 2 for Chapter 5. After solving the reaction-diffusion systems in Chapter 5 to simulate the propagation of electrochemical waves in human hearts, we expect to see how the propagation of electrochemical waves proceeds on the moving surface. Now, reduce this problem to solve the convection-diffusion equation (We will introduce this in Chapter 7.) on the moving domain. It is reasonable to solve the heat equation before solving the convection-diffusion equation. We solve the heat equation in Curvilinear coordinates in place of solving it in spherical or ellipsoid coordinates as Chapter 3 and Chapter 4 because the domain moves with time. And impose the explicit schemes on the solver because it is easy to find that we must extra define the values on the north and south poles head-on when using implicit schemes on it. Due to above reasons, we solve the heat equation in Curvilinear coordinates with the explicit schemes and check out this solver in Chapter 6. Therefore, we could solve the convection-diffusion equation in Chapter 7 with the discretization of Chapter 6. In the end, our future works are that combine the techniques in Chapter 7 with the reaction-diffusion systems whose domain approximates to human hearts. Expect to produce a primary and more real-life cardiac simulation of human beings. By the way, we are interested in observing if the total mass complies with the mass conservation law in each of our numerical solvers.

2、Numerical Scheme

2.1 Fast Fourier Transform

Before talking about the fast Fourier transform (FFT), we introduce the Fourier Transform and the discrete Fourier transform (DFT). We will use the FFT to discrete our spatial terms which is periodic in our equation. The contents in this section come from [14].

2.1.1 Fourier Transform

Transform $u(x)$ into Fourier series formula, then we get $u(x) = \sum_{k=0}^{n} a_k \phi_k(x)$, where a_k are constants, $\phi_k(x) = e^{ikx}$, $x \in [0,2\pi]$, $(f,g) = \int_0^{2\pi} f(x)\overline{g(x)}dx$ means inner product.

Theorem: **If** $f \in L^2(0,2\pi) = \left\{ f : (0,2\pi) \to \mathbb{C} \;\middle|\; \int_0^{2\pi} |f(x)|^2\, dx < \infty \right\}$,

$$\textbf{then } f(x) = \sum_{k=-\infty}^{\infty} \hat{f}_k e^{ikx}, \ \hat{f}_k = \frac{1}{2\pi}\int_0^{2\pi} f(x) e^{-ikx}\, dx. \tag{1}$$

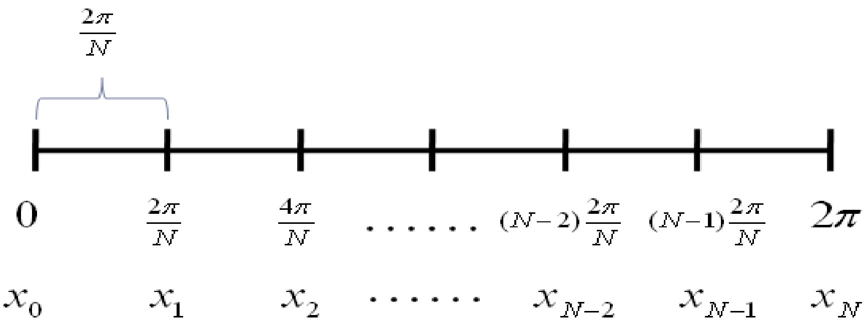
Here $\hat{f}_k$ is called continuous Fourier coefficient, and we have the properties:

$$\begin{cases} \mathbf{f(x)} = \sum\limits_{\mathbf{k=-\infty}}^{\infty} \hat{f}_k \mathbf{e^{ikx}} \\ \overline{\mathbf{f(x)}} = \sum\limits_{\mathbf{k=-\infty}}^{\infty} \bar{\hat{f}}_k \mathbf{e^{-ikx}} \end{cases} => \begin{cases} \mathbf{f(x)} = \sum\limits_{\mathbf{-k=-\infty}}^{\infty} \hat{f}_{-k} \mathbf{e^{-ikx}} \\ \overline{\mathbf{f(x)}} = \sum\limits_{\mathbf{k=-\infty}}^{\infty} \bar{\hat{f}}_k \mathbf{e^{-ikx}} \end{cases}$$

$$=> \hat{f}_{-k} = \bar{\hat{f}}_k \tag{2}$$

2.1.2 Discrete Fourier Transform

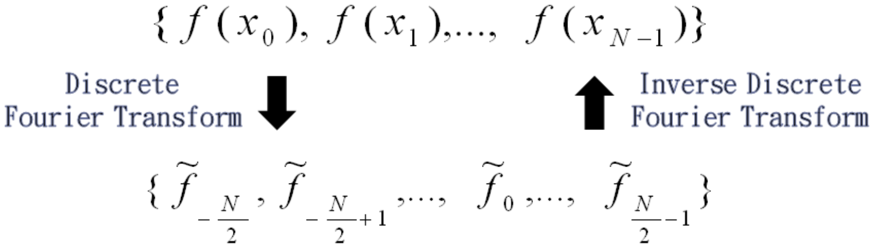
Discrete $(f,g) = \int_0^{2\pi} f(x)\overline{g(x)}dx$ by trapezoid rule and cut $[0,2\pi]$ into N partitions uniformly. Then derive $(f,g)_N = h\sum_{j=0}^{N-1} f(x_j)\overline{g(x_j)}$, where $h = \frac{2\pi}{N}$. Here mesh the grids as follows:



After using the same method as above on (1), we derive

$$f(x) \equiv \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \tilde{f}_k e^{ikx} \quad , \quad \tilde{f}_k = \frac{1}{N}\sum_{j=0}^{N-1} f(x_j)e^{-ikx_j}$$

where $x_j = jh = j\frac{2\pi}{N}$, $\tilde{f}_k$ is called discrete Fourier coefficient, f depends on N because f was cut into N partitions. So, call into being the formulation:



$$\{ f(x_0), f(x_1), \ldots, f(x_{N-1}) \}$$

$$\{ \tilde{f}_{-\frac{N}{2}}, \tilde{f}_{-\frac{N}{2}+1}, \ldots, \tilde{f}_0, \ldots, \tilde{f}_{\frac{N}{2}-1} \}$$

It is easy to know

$$f(x_j) = f_j = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \tilde{f}_k e^{ikx_j}, j = 1, 2, \ldots, N-1. \qquad (3)$$

In order to make the indices be the same with $0, 1, \ldots, N-1$, we show it in below contents:

Let

$$d_k \equiv \tilde{f}_{k-\frac{N}{2}} = \frac{1}{N}\sum_{j=0}^{N-1} f_j e^{-i(k-\frac{N}{2})x_j} = \frac{1}{N}\sum_{j=0}^{N-1} (f_j e^{i\frac{N}{2}x_j})e^{-ikx_j}$$

$$F_j \equiv f_j e^{i\frac{N}{2}x_j} \qquad (4)$$

We derived

$$d_k = \frac{1}{N}\sum_{j=0}^{N-1} (F_j)e^{-ikx_j}, \text{for } k = 0, 1, \ldots, N-1.$$

Substitute (3) into (4),

$$F_j \equiv f_j e^{i\frac{N}{2}x_j} = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \tilde{f}_k e^{i\left(k+\frac{N}{2}\right)x_j} = \sum_{k=0}^{N-1} \tilde{f}_{k-\frac{N}{2}} e^{ikx_j} = \sum_{k=0}^{N-1} d_k e^{ikx_j}, \text{for } j = 0, 1, \ldots, N-1.$$

Get the relationship between d_k and F_j,

$$\begin{cases} d_k = \frac{1}{N}\sum_{j=0}^{N-1} F_j e^{-ikx_j} & , \; k = 0,1,\dots,N-1. \\ F_j = \sum_{k=0}^{N-1} d_k e^{ikx_j} & , \; j = 0,1,\dots,N-1. \end{cases}$$

and the formulation:

$$\{F_0, F_1, \dots, F_{N-1}\} \xrightarrow[]{\text{Forward Fourier Transform}} \{d_0, d_1, \dots, d_{N-1}\}$$

$$\{d_0, d_1, \dots, d_{N-1}\} \xrightarrow[]{\text{Backward Fourier Transform}} \{F_0, F_1, \dots, F_{N-1}\}$$

In the configuration of matrix, the definition says

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{N-1} \end{bmatrix} = \frac{1}{N}\begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \cdots & \omega^{N-1} \\ \omega^0 & \omega^2 & \omega^4 & \cdots & \omega^{2(N-1)} \\ \omega^0 & \omega^3 & \omega^6 & \cdots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{N-1} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)^2} \end{bmatrix} \begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ \vdots \\ F_{N-1} \end{bmatrix} \tag{5}$$

where $\omega = e^{-i\frac{2\pi}{N}}$.

The $N \times N$ matrix in (5) is called the Fourier matrix,

$$H_N \equiv \frac{1}{N}\begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \cdots & \omega^{N-1} \\ \omega^0 & \omega^2 & \omega^4 & \cdots & \omega^{2(N-1)} \\ \omega^0 & \omega^3 & \omega^6 & \cdots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{N-1} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)^2} \end{bmatrix}$$

Here H_N is a symmetric matrix, and the Fourier matrix has an inverse matrix:

$$H_N{}^{-1} = \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(N-1)} \\ \omega^0 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(N-1)} \\ \omega^0 & \omega^{-3} & \omega^{-6} & \cdots & \omega^{-3(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{-(N-1)} & \omega^{-2(N-1)} & \cdots & \omega^{-(N-1)^2} \end{bmatrix}$$

Now, we have the forward Fourier transform

$$H_N \begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ \vdots \\ F_{N-1} \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{N-1} \end{bmatrix}$$

and the backward Fourier transform

$$H_N{}^{-1}\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{N-1} \end{bmatrix} = \begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ \vdots \\ F_{N-1} \end{bmatrix}, \text{where } H_N^{-1} = \overline{H}_N \,.$$

2.1.3 Fast Fourier Transform

The discrete Fourier transform that we used in the traditional way needs $O(N^2)$ operations. Cooley and Tukey [4] could achieve the DFT in $O(N \log N)$ operations in an algorithm which called the fast Fourier transform (FFT) which takes advantage of divide and conquer algorithm. We can write the DFT (5) as

$$d = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{N-1} \end{bmatrix} = \frac{1}{N} G_N \begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ \vdots \\ F_{N-1} \end{bmatrix} = \frac{1}{N} G_N F$$

where $$G_N = \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \cdots & \omega^{N-1} \\ \omega^0 & \omega^2 & \omega^4 & \cdots & \omega^{2(N-1)} \\ \omega^0 & \omega^3 & \omega^6 & \cdots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega^0 & \omega^{N-1} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)^2} \end{bmatrix}, F = \begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ \vdots \\ F_{N-1} \end{bmatrix}.$$

We will show how to compute $b = G_N F$ recursively. It needs to divide by N to complete the DFT because $d = \frac{b}{N}$. In the beginning, we show how the $n = N$ case works. FFT takes the advantages of G_N and $G_{\frac{N}{2}}$. Let $M = \frac{N}{2}$, and our indices of matrix are taken as Fortran form.

$b = G_N F$, $\omega_N = e^{-i\frac{2\pi}{N}}$ and suppose N is even here (If N is odd, then the process is similar with that we discuss here.). Then

$$b_j = (G_N F)_j = \sum_{k=0}^{N-1} \omega_N^{jk} F_k$$

$$= \sum_{\substack{k=0 \\ k\in\text{even}}}^{N-2} \omega_N^{jk} F_k + \sum_{\substack{k=1 \\ k\in\text{odd}}}^{N-1} \omega_N^{jk} F_k$$

Substitute $M = \frac{N}{2}$, $\ell = \frac{k}{2}$ and $\omega_M = e^{-i\frac{2\pi}{M}}$ into above equation.

$$b_j = \sum_{\ell=0}^{M-1} \omega_N^{j2\ell} F_{2\ell} + \sum_{\ell=0}^{M-1} \omega_N^{j(2\ell+1)} F_{2\ell+1}$$

$$= \sum_{\ell=0}^{M-1} \omega_M^{j\ell} F_{2\ell} + \omega_N^{j} \sum_{\ell=0}^{M-1} \omega_M^{j\ell} F_{2\ell+1}$$

In summary, the calculation of the DFT(N) can be performed by computing a pair of DFT($\frac{N}{2}$)s and some extra multiplications and additions. Consider n, ignore the $\frac{1}{n}$ temporarily, DFT(n) can be reduced to a pair of DFT($\frac{n}{2}$)s plus $O(n)$ extra operations which are multiplications and additions. We use above idea recursively and this is the most important techniques to make DFT solver more fast. Let n be a power of 2. Then the Fast Fourier Transform of size n can be accomplished in $n(2\log_2 n - 1) + 1$ additions and multiplications, plus a division by n [14]. We use the FFT to help computing our numerical solutions. And it leads to less computation by reducing $O(N^2)$ operations to $O(N \log_2 N)$ operations.

2.2 Second-Order Finite Difference Method

We will use Section 2.2.1 and Section 2.2.2 numerical methods to discrete our spatial terms of our equations.

2.2.1 Second-Order Central Difference Method

We refer to [14] to make this section complete. As below saying:

Let y be the function which depends on x. x_i is the interior mesh point, for $i = 1, 2, \ldots, N$. and $x_{i+1} - x_i = h$. Expanding y in a third Taylor polynomial at x_i to evaluate y at x_{i+1} and x_{i-1}, then we have

$$y(x_{i+1}) = y(x_i + h) = y(x_i) + hy'(x_i) + \frac{h^2}{2}y''(x_i) + \frac{h^3}{6}y'''(x_i) + \frac{h^4}{24}y^{(4)}\left(\xi_i^{+}\right) \quad (6)$$

for some ξ_i^{+} in (x_i, x_{i+1}), and

$$y(x_{i-1}) = y(x_i - h) = y(x_i) - hy'(x_i) + \frac{h^2}{2}y''(x_i) - \frac{h^3}{6}y'''(x_i) + \frac{h^4}{24}y^{(4)}(\xi_i^{-}) \quad (7)$$

for some ξ_i^- in (x_{i-1}, x_i), and assume $y \in C^4[x_{i-1}, x_{i+1}]$ here.

If subtract (7) from (6), then the terms involve $y''(x_i)$ and $y^{(4)}(x_i)$ are eliminated.

$$y'(x_i) = \frac{1}{2h}[y(x_{i+1}) - y(x_{i-1})] - \frac{h^2}{6}y'''(\eta_i)\,, \text{for some } \eta_i \text{ in } (x_{i-1}, x_{i+1}).$$

Therefore,

$$y'(x_i) \approx \frac{1}{2h}[y(x_{i+1}) - y(x_{i-1})] + O(h^2) \tag{8}$$

Use the same techniques, we could also have

$$y'(x_i) \approx \frac{1}{h}\left[y\left(x_{i+\frac{1}{2}}\right) - y\left(x_{i-\frac{1}{2}}\right)\right] + O(h^2) \tag{9}$$

If add (6) to (7), then the terms involve $y'(x_i)$ and $y'''(x_i)$ are eliminated.

$$y''(x_i) = \frac{1}{h^2}[y(x_{i+1}) - 2y(x_i) + y(x_{i-1})] - \frac{h^2}{24}\left(y^{(4)}(\xi_i^+) + y^{(4)}(\xi_i^-)\right).$$

The Intermediate Value Theorem can be used in above equation：

$$y''(x_i) = \frac{1}{h^2}[y(x_{i+1}) - 2y(x_i) + y(x_{i-1})] - \frac{h^2}{12}y^{(4)}(\xi_i), \text{for some } \xi_i \text{ in } (x_{i-1}, x_{i+1})$$

Therefore,

$$y''(x_i) \approx \frac{1}{h^2}[y(x_{i+1}) - 2y(x_i) + y(x_{i-1})] + O(h^2) \tag{10}$$

(8) (9) both are called central difference method for $y'(x_i)$.

(8) denotes central difference method 1 in our paper here.

(9) denotes central difference method 2 in our paper here.

(10) is called central difference formula for $y''(x_i)$.

They really have second-order accuracy.

2.2.2 Second-Order Symmetric Discretization

According to second-order accuracy of Section 2.2.1, we could produce a numerical method which was called Symmetric Discretization. This method builds up from the second-order central difference method. The progress of the symmetric discretization is as follows:

Step 1: We use the central difference method 2 for the whole $f(x_i)\frac{\partial g(x_i)}{\partial x}$ as Figure 1.

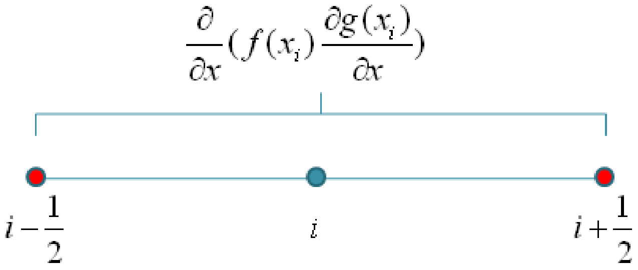

$\frac{\partial}{\partial x}(f(x_i)\frac{\partial g(x_i)}{\partial x})$
$i-\frac{1}{2}$
i
$i+\frac{1}{2}$

Figure 1

$$\frac{\partial}{\partial x}(f(x_i)\frac{\partial g(x_i)}{\partial x}) = \frac{f(x_{i+\frac{1}{2}})\frac{\partial g(x_{i+\frac{1}{2}})}{\partial x} - f(x_{i-\frac{1}{2}})\frac{\partial g(x_{i-\frac{1}{2}})}{\partial x}}{\Delta x}$$

Step 2: We use the central difference method 2 for $\frac{\partial g(x_{i+\frac{1}{2}})}{\partial x}$ and $\frac{\partial g(x_{i-\frac{1}{2}})}{\partial x}$ as Figure 2.

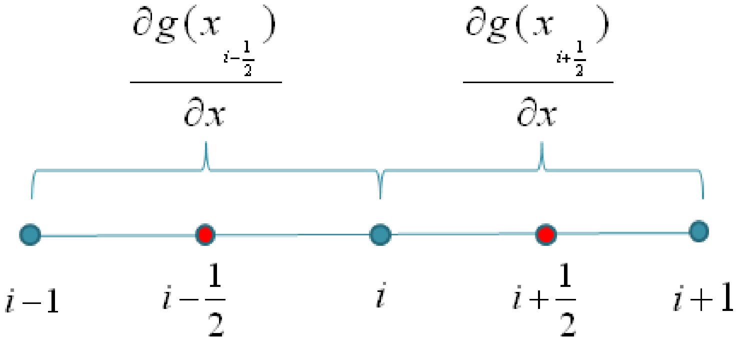

$\frac{\partial g(x_{i-\frac{1}{2}})}{\partial x}$
$\frac{\partial g(x_{i+\frac{1}{2}})}{\partial x}$
$i-1$
$i-\frac{1}{2}$
i
$i+\frac{1}{2}$
$i+1$

Figure 2

$$\frac{\partial g(x_{i+\frac{1}{2}})}{\partial x} = \frac{g(x_{i+1}) - g(x_i)}{\Delta x}, \frac{\partial g(x_{i-\frac{1}{2}})}{\partial x} = \frac{g(x_i) - g(x_{i-1})}{\Delta x}$$

Step 3: After substituting Step 2 into Step 1, we will get the symmetric discretization as follows.

$$\frac{\partial}{\partial x}(f(x_i)\frac{\partial g(x_i)}{\partial x}) = \frac{f(x)_{i+\frac{1}{2}}\frac{g(x_{i+1}) - g(x_i)}{\Delta x} - f(x)_{i-\frac{1}{2}}\frac{g(x_i) - g(x_{i-1})}{\Delta x}}{\Delta x}$$

Because of second-order accuracy in Step 1 and Step 2, we have second-order accuracy on the symmetric discretization.

2.3 Second-Order Crank-Nicolson Method

We will use the numerical methods in this section to discrete our spatiotemporal terms of our equations. The Crank-Nicolson method is unconditionally stable and has order of convergence $O(\Delta x^2 + \Delta t^2)$ can be found in Isaacson and Keller [9]. According to [5], when we integrating the o.d.e.

$$u'(t) = f(\,t\,, u(t))$$

on the interval $I_n = [t_n, t_{n+1}]$, we could have

$$u(t_{n+1}) - u(t_n) = \int_{t_n}^{t_{n+1}} f\big(s, u(s)\big)\, ds = I_n \tag{11}$$

Here we ignore spatial terms in u, i.e. the temporal terms dominate the accuracy. If we approximate the interval I_n, then we can compute $u(t_{n+1})$ from the old value $u(t_n)$. The trapezoid rule

$$I_n \approx \frac{\Delta t}{2}[f(t_n, u^n) + f(t_{n+1}, u^{n+1})]$$

is used to estimate the integral (11). Here $\Delta t = t_{n+1} - t_n$, $u^n = u(t_n)$. Consequently, we derive the formula of the Crank-Nicolson method:

$$\frac{u(t_{n+1}) - u(t_n)}{\Delta t} = \frac{[f(t_n, u^n) + f(t_{n+1}, u^{n+1})]}{2} \tag{12}$$

From (9), we have

$$u'\left(t_{n+\frac{1}{2}}\right) \approx \frac{u(t_{n+1}) - u(t_n)}{\Delta t} + O(h^2) \tag{13}$$

Take Taylor expansion for $f(t_n, u^n)$ and $f(t_{n+1}, u^{n+1})$ at $f(t_{n+\frac{1}{2}}, u^{n+\frac{1}{2}})$ and add each other similar to Section 2.2.1. Lastly, after combining with (13) we could have second-order accuracy on the Crank-Nicolson method.

2.4 Thomas Algorithm

In our numerical solvers, the most important procedure which saves our time and moves up our efficiency of computation is the method to solve the linear systems. Because the linear systems that we need to solve almost belong to tridiagonal systems, here we introduce the tridiagonal matrix algorithm (TDMA) which is known as the Thomas algorithm. The systems can be described as follows:

$$\begin{bmatrix} b_1 & c_1 & & . & 0 \\ a_2 & b_2 & c_2 & & . \\ & . & . & . & \\ . & & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & . & & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ . \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ . \\ d_{n-1} \\ d_n \end{bmatrix}$$

i.e. $$AX = B$$

There are n unknowns as X. The Thomas algorithm is a correction of the LU decomposition idea to solve linear systems with three bands diagonal coefficient matrix. We show our programming in our solver in MATLAB as follows:

```
%==============THOMAS ALGORITHM==================================
for k=2:n
        m=A(k,k-1)/A(k-1,k-1);
        A(k,k)=A(k,k)-m*A(k-1,k);
        B(k)=B(k)-m*B(k-1);
end
        X(M)=B(M)/A(M,M);
        k=n-1;
while k>=1
        X(k)=(X(k)-A(k,k+1)*X(k+1))/A(k,k);
        k=k-1;
end
%==============================================================
```

It is easy to know that the solution can be obtained in $0(n)$ operations instead of $0(n^3)$ by using the Gaussian elimination. After this section, we have enough numerical methods to proceed with our numerical solutions.

3、Fast Heat Solver in Spherical Coordinates

3.1 Heat Solver on a Spherical Surface Domain

3.1.1 Abstract

The solution of the heat equation on a spherical surface geometry has many applications in the areas of biology, geophysics, and engineering. We will present a simple and efficient method which is FFT fast direct solver for Heat equation on a spherical surface domain. These solvers on the truncated Fourier series expansion, and the Fourier coefficients are solved by second-order finite difference methods. Let the grids of the north and south poles shift half a grid away from the poles [12] (We will show the architecture of the grids later.). Next, combine the symmetry constraint of the Fourier coefficients in the solver (We need not use the symmetry constraint in the solver with the symmetric discretization.). Hence we could deal with the coordinate singularities easily without the conditions of the south and north poles. The process of FFT only takes $O(N\log_2 N)$ computations. After solving Fourier coefficients M times at the beginning and the end, the total operations of our solver only need $O(MN\log_2 N)$ arithmetic operations for $M \times N$ grid points. The total cost here does not include solving the linear systems of the unknown Fourier coefficients N times. We derive these unknowns by solving the linear systems whose matrices are $M \times M$ and tridiagonal N times by Thomas algorithm which costs $O(M)$ operations. If we impose (2) on our solver, then we only need to solve the linear systems $\frac{N}{2}$ times. Therefore, our solver needs $O(MN)$ arithmetic operations to solve these linear systems. All cost in our fast Heat solver needs only $O(MN\log_2 N)$ operations (M and N will be introduced later.). We will know that it is convenient to rewrite the equations in spherical and ellipsoid coordinates in Chapter 3 and Chapter 4, respectively.

3.1.2 Solvers with the Central Difference Method

According to [6], we have the formula of the heat equation on a spherical surface domain in spherical coordinates as

$$\begin{cases} \mathrm{u_t} = \Delta_s \mathrm{u} \\ \Delta_s \mathrm{u} = \nabla_s^2 \mathrm{u} = \dfrac{1}{\mathrm{r}^2}\left(\dfrac{\partial^2 \mathrm{u}}{\partial \phi^2} + \cot(\phi)\dfrac{\partial \mathrm{u}}{\partial \phi} + \dfrac{1}{\sin^2(\phi)}\dfrac{\partial^2 \mathrm{u}}{\partial \theta^2}\right) \end{cases} \tag{14}$$

We also can derive the surface Laplacian term of (14) by [10] and set radius r be a constant R at the same time or by [16]

$$\nabla_s^2 \mathrm{u} = \nabla^2 \mathrm{u} - \frac{\partial^2 \mathrm{u}}{\partial \mathbf{n}^2} - \kappa \frac{\partial \mathrm{u}}{\partial \mathbf{n}} \tag{15}$$

Here $\mathbf{n} = \mathrm{r}$: normal, $\kappa = \frac{2}{\mathrm{r}}$: mean curvature, $\mathrm{x = R\sin\phi\cos\theta}$, $\mathrm{y = R\sin\phi\sin\theta, z = R\cos\phi}$ in spherical coordinates, and

$$\Delta \mathrm{u} = \nabla^2 \mathrm{u} = \frac{\partial^2 \mathrm{u}}{\partial \mathrm{r}^2} + \frac{2}{\mathrm{r}}\frac{\partial \mathrm{u}}{\partial \mathrm{r}} + \frac{1}{\mathrm{r}^2}\left(\frac{\partial^2 \mathrm{u}}{\partial \phi^2} + \cot(\phi)\frac{\partial \mathrm{u}}{\partial \phi} + \frac{1}{\sin^2(\phi)}\frac{\partial^2 \mathrm{u}}{\partial \theta^2}\right) \tag{16}$$

comes from [10]. After replacing $\mathrm{n = r}$ and $\kappa = \frac{2}{\mathrm{r}}$ in (15) and substituting (16) into (15). In (14), ignore r by setting r a constant R, and denote $\mathrm{u(R, \phi, \theta, t) \approx u(\phi, \theta, t)}$.

$0 \leq \phi < \pi$ represent the directions of latitude,

$0 \leq \theta < 2\pi$ represent the directions of longitude.

(14) was discreted by cutting the space as the following uniform grids which shift half a grid away from the poles.

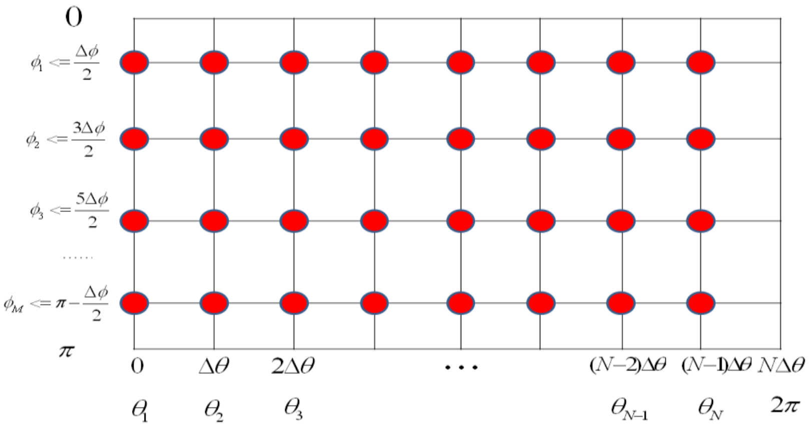


Figure 3

Given any $N = 2M$, $\Delta\phi = \frac{\pi}{M} = \Delta\theta = \frac{2\pi}{N}$,

$$\phi_i = (i - 0.5)\Delta\phi, i = 1, \dots, M.$$

$$\theta_j = j\Delta\theta, j = 1, \dots, N.$$

We avoid putting the grids directly at the poles because we hope that no conditions in the poles are needed [12].

Step 1: We deal with the equation in spatiotemporal organization by the Crank-Nicolson method as (12)

$$u_t = \Delta_s u$$

$$=> \frac{u^{n+1} - u^n}{\Delta t} = \frac{\Delta_s u^{n+1} + \Delta_s u^n}{2}$$

$$=> \Delta_s u^{n+1} - \frac{2}{\Delta t} u^{n+1} = -\Delta_s u^n - \frac{2}{\Delta t} u^n \tag{17}$$

Step 2: We perform fast Fourier transform to θ. Since the solution u on a spherical surface domain is periodic in θ, we can approximate it by the truncated Fourier series as [10].

$$u(\phi, \theta) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} u_k(\phi)\, e^{ik\theta} \tag{18}$$

where

$$u_k(\phi) = \frac{1}{N} \sum_{j=0}^{N-1} u(\phi, \theta) e^{-ik\theta_j}$$

Substitute (18) into $\Delta_s u$ as (14),

$$\Delta_s u = \frac{1}{R^2}\left(\frac{\partial^2 \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} u_k(\phi)\, e^{ik\theta}}{\partial\phi^2} + \cot(\phi) \frac{\partial \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} u_k(\phi)\, e^{ik\theta}}{\partial\phi} + \frac{1}{\sin^2(\phi)} \frac{\partial^2 \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} u_k(\phi)\, e^{ik\theta}}{\partial\theta^2} \right)$$

$$= \frac{1}{R^2}\left(\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \frac{\partial^2 u_k(\phi) e^{ik\theta}}{\partial\phi^2} + \cot(\phi) \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \frac{\partial u_k(\phi) e^{ik\theta}}{\partial\phi} + \frac{1}{\sin^2(\phi)} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \frac{\partial^2 u_k(\phi) e^{ik\theta}}{\partial\theta^2} \right)$$

$$
=\frac{1}{R^2}\left(\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\frac{\partial^2 u_k(\phi)}{\partial\phi^2}e^{ik\theta}+\cot(\phi)\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\frac{\partial u_k(\phi)}{\partial\phi}e^{ik\theta}+\frac{1}{\sin^2(\phi)}\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}u_k(\phi)(-k^2)e^{ik\theta}\right)
$$

$$
=\frac{1}{R}\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\left(\frac{\partial^2 u_k(\phi)}{\partial\phi^2}+\cot(\phi)\frac{\partial u_k(\phi)}{\partial\phi}-\frac{k^2}{\sin^2(\phi)}u_k(\phi)\right)e^{ik\theta} \qquad (19)
$$

Substitute the surface Laplacian term of (14) into (17),

$$
\frac{1}{R^2}\left(\frac{\partial^2 u^{n+1}}{\partial\phi^2}+\cot(\phi)\frac{\partial u^{n+1}}{\partial\phi}+\frac{1}{\sin^2(\phi)}\frac{\partial^2 u^{n+1}}{\partial\theta^2}\right)-\frac{2}{\Delta t}u^{n+1}
$$

$$
=-\frac{1}{R^2}\left(\frac{\partial^2 u^{n}}{\partial\phi^2}+\cot(\phi)\frac{\partial u^{n}}{\partial\phi}+\frac{1}{\sin^2(\phi)}\frac{\partial^2 u^{n}}{\partial\theta^2}\right)-\frac{2}{\Delta t}u^{n} \qquad (20)
$$

Substitute (18) into (20), and from (19) we can derive

$$
\frac{1}{R^2}\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\left(\frac{\partial^2 u_k^{n+1}(\phi)}{\partial\phi^2}+\cot(\phi)\frac{\partial u_k^{n+1}(\phi)}{\partial\phi}-\frac{k^2}{\sin^2(\phi)}u_k^{n+1}(\phi)\right)e^{ik\theta}-\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\frac{2}{\Delta t}u_k^{n+1}(\phi)\,e^{ik\theta}
$$

$$
=-\frac{1}{R^2}\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\left(\frac{\partial^2 u_k^{n}(\phi)}{\partial\phi^2}+\cot(\phi)\frac{\partial u_k^{n}(\phi)}{\partial\phi}-\frac{k^2}{\sin^2(\phi)}u_k^{n}(\phi)\right)e^{ik\theta}-\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\frac{2}{\Delta t}u_k^{n}(\phi)\,e^{ik\theta}
$$

$$
=>\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\left(\frac{1}{R^2}(\frac{\partial^2 u_k^{n+1}(\phi)}{\partial\phi^2}+\cot(\phi)\frac{\partial u_k^{n+1}(\phi)}{\partial\phi}-\frac{k^2}{\sin^2(\phi)}u_k^{n+1}(\phi))-\frac{2}{\Delta t}u_k^{n+1}(\phi)\right)e^{ik\theta}
$$

$$
=\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}(-\frac{1}{R^2}(\frac{\partial^2 u_k^{n}(\phi)}{\partial\phi^2}+\cot(\phi)\frac{\partial u_k^{n}(\phi)}{\partial\phi}-\frac{k^2}{\sin^2(\phi)}u_k^{n}(\phi))-\frac{2}{\Delta t}u_k^{n}(\phi))\,e^{ik\theta}
$$

Here the superscript of $u_k^{n+1}(\phi)$ means the $(n+1)$-th time step, and the subscript of $u_k^{n+1}(\phi)$ means the k-th Fourier coefficient.

Eliminate $\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}$ and $e^{ik\theta}$ from both sides of the equation,

$$
\frac{1}{R^2}(\frac{\partial^2 u_k^{n+1}(\phi)}{\partial\phi^2}+\cot(\phi)\frac{\partial u_k^{n+1}(\phi)}{\partial\phi}-\frac{k^2}{\sin^2(\phi)}u_k^{n+1}(\phi))-\frac{2}{\Delta t}u_k^{n+1}(\phi)
$$

$$
=-\frac{1}{R^2}(\frac{\partial^2 u_k^{n}(\phi)}{\partial\phi^2}+\cot(\phi)\frac{\partial u_k^{n}(\phi)}{\partial\phi}-\frac{k^2}{\sin^2(\phi)}u_k^{n}(\phi))-\frac{2}{\Delta t}u_k^{n}(\phi)
$$

After multiplying by R^2 to both sides of the equation and equaling the Fourier coefficients, $u_k(\phi)$ satisfies

$$\frac{\partial^2 u_k^{n+1}(\phi)}{\partial\phi^2}+\cot(\phi)\frac{\partial u_k^{n+1}(\phi)}{\partial\phi}-\frac{k^2}{\sin^2(\phi)}u_k^{n+1}(\phi)-\frac{2R^2}{\Delta t}u_k^{n+1}(\phi)$$
$$=-\frac{\partial^2 u_k^{n}(\phi)}{\partial\phi^2}-\cot(\phi)\frac{\partial u_k^{n}(\phi)}{\partial\phi}+\frac{k^2}{\sin^2(\phi)}u_k^{n}(\phi)-\frac{2R^2}{\Delta t}u_k^{n}(\phi)$$

The Step 2 is equal to do FFT in θ as follows：

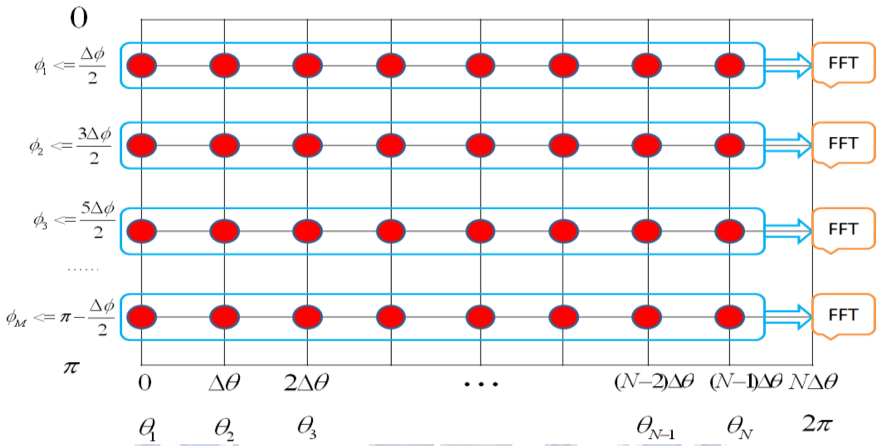


Figure 4

Step 3: We use the second-order central difference method for ϕ, i.e.

$$\frac{\partial U_i}{\partial\phi}\approx\frac{U_{i+1}-U_{i-1}}{2\Delta\phi}\ \text{and}\ \frac{\partial^2 U_i}{\partial\phi^2}\approx\frac{U_{i+1}-2U_i+U_{i-1}}{\Delta\phi^2}$$

For each Fourier modes k $(k=-\frac{N}{2},-\frac{N}{2}+1,\dots,0,1,\dots,\frac{N}{2}-1.)$, we set

$$u_k^{n+1}(\phi)=U^{n+1}\ \text{at}\ T=(n+1)\Delta t$$

$$u_k^{n}(\phi)=U^{n}\ \text{at}\ T=n\Delta t$$

and solve the following equations.

$$\frac{U_{i+1}^{n+1}-2U_i^{n+1}+U_{i-1}^{n+1}}{\Delta\phi^2}+\cot(\phi_i)\frac{U_{i+1}^{n+1}-U_{i-1}^{n+1}}{2\Delta\phi}-\frac{k^2}{\sin^2(\phi_i)}U_i^{n+1}-\frac{2R^2}{\Delta t}U_i^{n+1}$$
$$=-\frac{U_{i+1}^{n}-2U_i^{n}+U_{i-1}^{n}}{\Delta\phi^2}-\cot(\phi_i)\frac{U_{i+1}^{n}-U_{i-1}^{n}}{2\Delta\phi}+\frac{k^2}{\sin^2(\phi_i)}U_i^{n}-\frac{2R^2}{\Delta t}U_i^{n}$$

for $i=1,\dots,M.$

Both sides of the equations were multiplied by $\Delta\phi^2$ and rearrange them,

$$(1-\frac{\cot(\phi_i)\Delta\phi}{2})U_{i-1}^{n+1}+\left(-2-\frac{k^2\Delta\phi^2}{\sin^2(\phi_i)}-\frac{2R^2\Delta\phi^2}{\Delta t}\right)U_i^{n+1}+(1+\frac{\cot(\phi_i)\Delta\phi}{2})U_{i+1}^{n+1}$$
$$=-(1-\frac{\cot(\phi_i)\Delta\phi}{2})U_{i-1}^{n}+\left(2+\frac{k^2\Delta\phi^2}{\sin^2(\phi_i)}-\frac{2R^2\Delta\phi^2}{\Delta t}\right)U_i^{n}-(1+\frac{\cot(\phi_i)\Delta\phi}{2})U_{i+1}^{n}.\quad(21)$$

Therefore, we could set that

$$A_i = 1 - \frac{\cot(\phi_i)\Delta\phi}{2}$$

$$B_i = -2 - \frac{k^2\Delta\phi^2}{\sin^2(\phi_i)} - \frac{2R^2\Delta\phi^2}{\Delta t}$$

$$C_i = 1 + \frac{\cot(\phi_i)\Delta\phi}{2}$$

$$D_i = -B_i - \frac{4R^2\Delta\phi^2}{\Delta t}$$

for $i = 1, \dots, M$. There are M+2 unknowns, i.e. $U_0, \dots, U_{M+1}$, but we only have M equations. Before solving the governing equation in spherical coordinates as (14), we burned with curiosity over that the Fourier coefficients of a function in the coordinates satisfy the symmetry constrain [12][13]. Consider the transformation between Cartesian coordinate systems and spherical coordinates. Because $x = R\sin\phi\cos\theta$, $y = R\sin\phi\sin\theta$, $z = R\cos\phi$, we could replace ϕ by $-\phi$ and θ by $\theta + \pi$ to derive that the Cartesian coordinates of points on a spherical surface is unchanged.

$$u(-\phi, \theta) = u(\phi, \theta + \pi)$$

Using above equalities, and

$$\begin{cases} u(-\phi, \theta) = \sum_{k=-\infty}^{\infty} u_k(-\phi)e^{ik\theta} \\ u(\phi, \theta + \pi) = \sum_{k=-\infty}^{\infty} u_k(\phi)e^{ik(\theta+\pi)} = \sum_{k=-\infty}^{\infty} (-1)^k u_k(\phi)e^{ik\theta} \end{cases}$$

Lead to

$$\sum_{k=-\infty}^{\infty} u_k(-\phi)e^{ik\theta} = \sum_{k=-\infty}^{\infty} (-1)^k u_k(\phi)e^{ik\theta}$$

Therefore, the k-th Fourier coefficient satisfies

$$u_k(-\phi) = (-1)^k u_k(\phi)$$

Use above equalities and below property which the equation (14) is inherent in

$$u(\phi, \theta) = u(2\pi + \phi, \theta), \forall\phi$$

Consequently, we could get

$$\mathrm{u}(\phi-\pi,\theta)=\sum_{k=-\infty}^{\infty}\mathrm{u}_{k}(\phi-\pi)e^{ik\theta}=\sum_{k=-\infty}^{\infty}\mathrm{u}_{k}(-(\pi-\phi))e^{ik\theta}$$

$$=\sum_{k=-\infty}^{\infty}(-1)^{k}\mathrm{u}_{k}(\pi-\phi)e^{ik\theta}$$

and $$\mathrm{u}(\phi-\pi,\theta)=\mathrm{u}(2\pi+(\phi-\pi),\theta)=\mathrm{u}(\pi+\phi,\theta)$$

$$=\sum_{k=-\infty}^{\infty}\mathrm{u}_{k}(\pi+\phi)e^{ik\theta}$$

Lead to

$$\sum_{k=-\infty}^{\infty}(-1)^{k}\mathrm{u}_{k}(\phi-\pi)e^{ik\theta}=\sum_{k=-\infty}^{\infty}\mathrm{u}_{k}(\pi+\phi)e^{ik\theta}$$

Thus we have the k-th Fourier coefficient of $\mathrm{u}(\pi+\phi,\theta)$ satisfies

$$\mathrm{u}_{k}(\pi+\phi)=(-1)^{k}\mathrm{u}_{k}(\pi-\phi)$$

Take advantage of these symmetry constraints to help solving the tridiagonal linear systems for U_i in (21), $i=1,\dots,M$, by deriving the boundary values for U_0 and U_{M+1}. We solve these unknowns by the Thomas algorithm with $O(M)$ arithmetic operations. Because of the symmetry constraints, U_0 and U_{M+1} can be obtained by

$$\mathrm{U}_0=\mathrm{u}_k(\phi_0)=\mathrm{u}_k\left(-\frac{\Delta\phi}{2}\right)=(-1)^k\mathrm{u}_k\left(\frac{\Delta\phi}{2}\right)=(-1)^k\mathrm{U}_1$$

$$\mathrm{U}_{M+1}=\mathrm{u}_k(\phi_{M+1})=\mathrm{u}_k\left(\pi+\frac{\Delta\phi}{2}\right)=(-1)^k\mathrm{u}_k\left(\pi-\frac{\Delta\phi}{2}\right)=(-1)^k\mathrm{U}_M$$

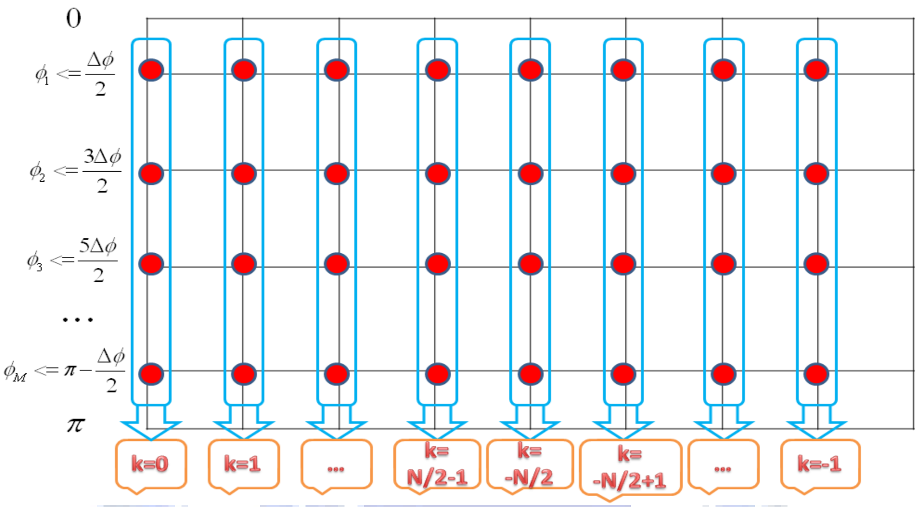
for each k-th Fourier coefficient. Then we could produce the following matrices,

$$\begin{bmatrix}(-1)^k A_1+B_1 & C_1 & & & \\ A_2 & B_2 & C_2 & & \\ & \cdot & \cdot & \cdot & \\ & & A_{M-1} & B_{M-1} & C_{M-1} \\ & & & A_M & B_M+(-1)^k C_M\end{bmatrix}\begin{bmatrix}U_1^{n+1}\\ U_2^{n+1}\\ \cdot\\ U_{M-1}^{n+1}\\ U_M^{n+1}\end{bmatrix}$$

$$
= \begin{bmatrix}
(-1)^{k+1}A_1 + D_1 & -C_1 & & & & \\
-A_2 & D_2 & -C_2 & & & \\
 & \cdot & \cdot & \cdot & & \\
 & & -A_{M-1} & D_{M-1} & & -C_{M-1} \\
 & & & -A_M & & D_M + (-1)^{k+1}C_M
\end{bmatrix}
\begin{bmatrix}
U_1^n \\ U_2^n \\ \cdot \\ U_{M-1}^n \\ U_M^n
\end{bmatrix}
$$

and we must solve these linear systems N times.

The Step 3 is equal to solve the unknowns in each column of the linear systems as follows,



0
$\phi_1 <= \frac{\Delta\phi}{2}$
$\phi_2 <= \frac{3\Delta\phi}{2}$
$\phi_3 <= \frac{5\Delta\phi}{2}$
...
$\phi_M <= \pi - \frac{\Delta\phi}{2}$
π
k=0
k=1
...
k=N/2-1
k=-N/2
k=-N/2+1
...
k=-1

Figure 5: The order of n here we follow the FFT in MATLAB.

Step 4: We perform inverse fast Fourier transform (IFFT) in θ, After solving the matrices, we do IFFT in each row on our grids. The Step 4 is equal to do IFFT in θ as follows：

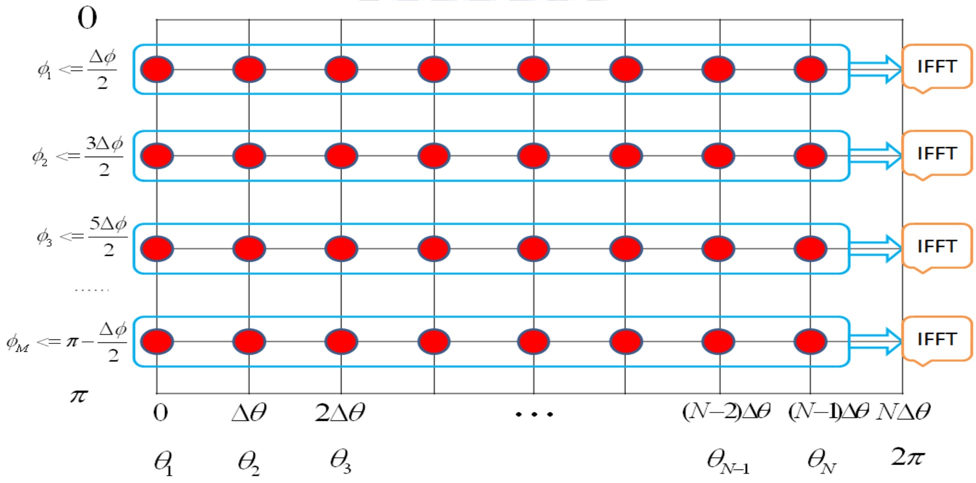


Figure 6

According to the procedure, i.e. Step 1=>Step 2=>Step 3=>Step 4, we could complete the numerical solution at one time step and we can derive the numerical solution at any time by keeping performing the procedure recursively.

3.1.3 Solvers with the Symmetric Discretization

According to [6], we have the formula of the heat equation on a spherical surface domain in spherical coordinates as

$$\begin{cases} \mathrm{u_t} = \Delta_{\mathrm{s}} \mathrm{u} \\ \Delta_{\mathrm{s}} \mathrm{u} = \nabla_{\mathrm{s}}^2 \mathrm{u} = \dfrac{1}{\mathrm{r}^2}\left(\dfrac{1}{\sin\phi}\dfrac{\partial}{\partial\phi}(\sin\phi\dfrac{\partial \mathrm{u}}{\partial\phi}) + \dfrac{1}{\sin^2(\phi)}\dfrac{\partial^2 \mathrm{u}}{\partial\theta^2}\right) \end{cases} \tag{22}$$

Here r is a constant R. We also can derive the surface Laplacian term of (22) and the settings of the grids as Section 3.1.2 says.

Step 1: Similar to Section 3.1.2 Step 1.

Step 2: Similar to Section 3.1.2 Step 2 and substitute (18) into (22),

$$\Delta_{\mathrm{s}} \mathrm{u} = \frac{1}{\mathrm{R}^2}\left(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}(\sin\phi\frac{\partial\sum_{\mathrm{k}=-\frac{\mathrm{N}}{2}}^{\frac{\mathrm{N}}{2}-1}\mathrm{u_k}(\phi)\,\mathrm{e}^{\mathrm{i}\mathrm{k}\theta}}{\partial\phi}) + \frac{1}{\sin^2(\phi)}\frac{\partial^2\sum_{\mathrm{k}=-\frac{\mathrm{N}}{2}}^{\frac{\mathrm{N}}{2}-1}\mathrm{u_k}(\phi)\,\mathrm{e}^{\mathrm{i}\mathrm{k}\theta}}{\partial\theta^2}\right)$$

$$= \frac{1}{\mathrm{R}^2}\left(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}(\sin\phi\sum_{\mathrm{k}=-\frac{\mathrm{N}}{2}}^{\frac{\mathrm{N}}{2}-1}\frac{\partial \mathrm{u_k}(\phi)\mathrm{e}^{\mathrm{i}\mathrm{k}\theta}}{\partial\phi}) + \frac{1}{\sin^2(\phi)}\sum_{\mathrm{k}=-\frac{\mathrm{N}}{2}}^{\frac{\mathrm{N}}{2}-1}\frac{\partial^2 \mathrm{u_k}(\phi)\mathrm{e}^{\mathrm{i}\mathrm{k}\theta}}{\partial\theta^2}\right)$$

$$= \frac{1}{\mathrm{R}^2}\left(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}(\sum_{\mathrm{k}=-\frac{\mathrm{N}}{2}}^{\frac{\mathrm{N}}{2}-1}\sin\phi\frac{\partial \mathrm{u_k}(\phi)}{\partial\phi}\mathrm{e}^{\mathrm{i}\mathrm{k}\theta}) + \sum_{\mathrm{k}=-\frac{\mathrm{N}}{2}}^{\frac{\mathrm{N}}{2}-1}\frac{1}{\sin^2(\phi)}\mathrm{u_k}(\phi)(-\mathrm{k}^2)\mathrm{e}^{\mathrm{i}\mathrm{k}\theta}\right)$$

$$= \frac{1}{\mathrm{R}^2}\sum_{\mathrm{k}=-\frac{\mathrm{N}}{2}}^{\frac{\mathrm{N}}{2}-1}\left(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}(\sin\phi\frac{\partial \mathrm{u_k}(\phi)}{\partial\phi}) - \frac{\mathrm{k}^2}{\sin^2(\phi)}\mathrm{u_k}(\phi)\right)\mathrm{e}^{\mathrm{i}\mathrm{k}\theta} \tag{23}$$

Substitute the surface Laplacian term of (22) into (17),

$$\frac{1}{\mathrm{R}^2}(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}(\sin\phi\frac{\partial \mathrm{u}^{\mathrm{n}+1}}{\partial\phi}) + \frac{1}{\sin^2(\phi)}\frac{\partial^2 \mathrm{u}^{\mathrm{n}+1}}{\partial\theta^2}) - \frac{2}{\Delta \mathrm{t}}\mathrm{u}^{\mathrm{n}+1}$$

$$= -\frac{1}{R^2}\left(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}\left(\sin\phi\frac{\partial u^n}{\partial\phi}\right)+\frac{1}{\sin^2(\phi)}\frac{\partial^2 u^n}{\partial\theta^2}\right)-\frac{2}{\Delta t}u^n \qquad (24)$$

Substitute (18) into (24), and from (23) we can derive

$$\frac{1}{R^2}\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\left(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}(\sin\phi\frac{\partial u_k^{n+1}(\phi)}{\partial\phi})-\frac{k^2}{\sin^2(\phi)}u_k^{n+1}(\phi)\right)e^{ik\theta}-\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\frac{2}{\Delta t}u_k^{n+1}(\phi)\,e^{ik\theta}$$

$$= -\frac{1}{R^2}\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\left(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}(\sin\phi\frac{\partial u_k^{n}(\phi)}{\partial\phi})-\frac{k^2}{\sin^2(\phi)}u_k^{n}(\phi)\right)e^{ik\theta}-\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\frac{2}{\Delta t}u_k^{n}(\phi)\,e^{ik\theta}$$

$$=> \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\left(\frac{1}{R^2}\left(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}(\sin\phi\frac{\partial u_k^{n+1}(\phi)}{\partial\phi})-\frac{k^2}{\sin^2(\phi)}u_k^{n+1}(\phi)\right)-\frac{2}{\Delta t}u_k^{n+1}(\phi)\right)e^{ik\theta}$$

$$= \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\left(-\frac{1}{R^2}\left(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}(\sin\phi\frac{\partial u_k^{n}(\phi)}{\partial\phi})-\frac{k^2}{\sin^2(\phi)}u_k^{n}(\phi)\right)-\frac{2}{\Delta t}u_k^{n}(\phi)\right)e^{ik\theta}$$

Eliminate $\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}$ and $e^{ik\theta}$ from both sides of the equation,

$$\frac{1}{R^2}(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}(\sin\phi\frac{\partial u_k^{n+1}(\phi)}{\partial\phi})-\frac{k^2}{\sin^2(\phi)}u_k^{n+1}(\phi))-\frac{2}{\Delta t}u_k^{n+1}(\phi)$$
$$= -\frac{1}{R^2}(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}(\sin\phi\frac{\partial u_k^{n}(\phi)}{\partial\phi})-\frac{k^2}{\sin^2(\phi)}u_k^{n}(\phi))-\frac{2}{\Delta t}u_k^{n}(\phi)$$

After multiplying by R^2 to both sides of the equation and equaling the Fourier coefficients, $u_k(\phi)$ satisfies

$$\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}(\sin\phi\frac{\partial u_k^{n+1}(\phi)}{\partial\phi})-\frac{k^2}{\sin^2(\phi)}u_k^{n+1}(\phi)-\frac{2R^2}{\Delta t}u_k^{n+1}(\phi)$$
$$= -\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}\left(\sin\phi\frac{\partial u_k^{n}(\phi)}{\partial\phi}\right)+\frac{k^2}{\sin^2(\phi)}u_k^{n}(\phi)-\frac{2R^2}{\Delta t}u_k^{n}(\phi)$$

The Step 2 is equal to do FFT for θ as Figure 4.

Step 3: We use the symmetric discretization for ϕ, i.e.

$$\frac{\partial}{\partial\phi}(\sin(\phi_i)\frac{\partial U_i}{\partial\phi})\approx\frac{\sin\phi_{i+\frac{1}{2}}(U_{i+1}-U_i)-\sin\phi_{i-\frac{1}{2}}(U_i-U_{i-1})}{\Delta\phi^2}$$

For each Fourier modes k ($k = -\frac{N}{2}, -\frac{N}{2}+1, \dots, 0, 1, \dots, \frac{N}{2}-1.$), we set

$$u_k^{n+1}(\phi) = U^{n+1} \text{ at time } (n+1)\Delta t$$

$$u_k^{n}(\phi) = U^{n} \text{ at time } n\Delta t$$

and solve the following equations.

$$\frac{1}{\sin\phi_i}\frac{\sin\phi_{i+\frac{1}{2}}(U_{i+1}^{n+1} - U_i^{n+1}) - \sin\phi_{i-\frac{1}{2}}(U_i^{n+1} - U_{i-1}^{n+1})}{\Delta\phi^2} - \frac{k^2}{\sin^2(\phi_i)}U_i^{n+1} - \frac{2R^2}{\Delta t}U_i^{n+1}$$

$$= -\frac{1}{\sin\phi_i}\frac{\sin\phi_{i+\frac{1}{2}}(U_{i+1}^{n} - U_i^{n}) - \sin\phi_{i-\frac{1}{2}}(U_i^{n} - U_{i-1}^{n})}{\Delta\phi^2} + \frac{k^2}{\sin^2(\phi_i)}U_i^{n} - \frac{2R^2}{\Delta t}U_i^{n}$$

for $i = 1, \dots, M$.

Both sides of the equations were multiplied by $\Delta\phi^2\sin(\phi_i)$ and rearrange them,

$$\sin\phi_{i-\frac{1}{2}}U_{i-1}^{n+1} + \left(-\sin\phi_{i-\frac{1}{2}} - \sin\phi_{i+\frac{1}{2}} - \frac{k^2\Delta\phi^2}{\sin(\phi_i)} - \frac{2R^2\Delta\phi^2\sin(\phi_i)}{\Delta t}\right)U_i^{n+1} + \sin\phi_{i+\frac{1}{2}}U_{i+1}^{n+1}$$

$$= -\sin\phi_{i-\frac{1}{2}}U_{i-1}^{n} + \left(\sin\phi_{i-\frac{1}{2}} + \sin\phi_{i+\frac{1}{2}} + \frac{k^2\Delta\phi^2}{\sin(\phi_i)} - \frac{2R^2\Delta\phi^2\sin(\phi_i)}{\Delta t}\right)U_i^{n} - \sin\phi_{i+\frac{1}{2}}U_{i+1}^{n} \quad (25)$$

Therefore, we could set that

$$A_i = \sin\phi_{i-\frac{1}{2}}$$

$$B_i = -\sin\phi_{i-\frac{1}{2}} - \sin\phi_{i+\frac{1}{2}} - \frac{k^2\Delta\phi^2}{\sin(\phi_i)} - \frac{2R^2\Delta\phi^2\sin(\phi_i)}{\Delta t}$$

$$C_i = \sin\phi_{i+\frac{1}{2}}$$

$$D_i = -B_i - \frac{4R^2\Delta\phi^2\sin(\phi_i)}{\Delta t}$$

for $i = 1, \dots, M$.

Because the coefficient of U_0 is $\sin\phi_{\frac{1}{2}} = 0$ and the coefficient of U_{M+1} is $\sin\phi_{M+\frac{1}{2}} = 0$, we have $A_1 = 0$ and $C_M = 0$. Therefore, we do not need symmetric constrain which is a great news for us to produce the following matrices,

$$\begin{bmatrix} B_1 & C_1 & & & \\ A_2 & B_2 & C_2 & & \\ & . & . & . & \\ & & A_{M-1} & B_{M-1} & C_{M-1} \\ & & & A_M & B_M \end{bmatrix} \begin{bmatrix} U_1^{n+1} \\ U_2^{n+1} \\ . \\ U_{M-1}^{n+1} \\ U_M^{n+1} \end{bmatrix} = \begin{bmatrix} D_1 & -C_1 & & & \\ -A_2 & D_2 & -C_2 & & \\ & . & . & . & \\ & & -A_{M-1} & D_{M-1} & -C_{M-1} \\ & & & -A_M & D_M \end{bmatrix} \begin{bmatrix} U_1^{n} \\ U_2^{n} \\ . \\ U_{M-1}^{n} \\ U_M^{n} \end{bmatrix}$$

Because $\mathrm{A}_{\mathrm{i}+1} = \sin\phi_{\mathrm{i}+\frac{1}{2}} = \mathrm{C}_{\mathrm{i}}$, for $\mathrm{i} = 1, \ldots, \mathrm{M}-1$, the matrices will become symmetric matrices and lead to less computations. It reduces above matrices to be

$$\begin{bmatrix} B_1 & C_1 & & & \\ C_1 & B_2 & C_2 & & \\ & . & . & . & \\ & & C_{M-2} & B_{M-1} & C_{M-1} \\ & & & C_{M-1} & B_M \end{bmatrix} \begin{bmatrix} U_1^{n+1} \\ U_2^{n+1} \\ . \\ U_{M-1}^{n+1} \\ U_M^{n+1} \end{bmatrix} = \begin{bmatrix} D_1 & -C_1 & & & \\ -C_1 & D_2 & -C_2 & & \\ & . & . & . & \\ & & -C_{M-2} & D_{M-1} & -C_{M-1} \\ & & & -C_{M-1} & D_M \end{bmatrix} \begin{bmatrix} U_1^{n} \\ U_2^{n} \\ . \\ U_{M-1}^{n} \\ U_M^{n} \end{bmatrix}$$

And these linear systems need to be solved N times by the Thomas algorithm here. The Step 3 is equal to solve the unknowns in the linear systems in each Fourier mode as Figure 5.

Step 4: Similar to Section 3.1.2 Step 4.

According to the procedure, i.e. Step 1=>Step 2=>Step 3=>Step 4, we could complete the numerical solution at one time step and we can derive the numerical solution at any time by keeping performing the procedure recursively.

3.2 Numerical Results

We provide the exact solution

$$u = e^{-2\frac{t}{R^2}}(x + y + z) = e^{-2\frac{t}{R^2}}(R\sin(\phi)\cos(\theta) + R\sin(\phi)\sin(\theta) + R\cos(\phi)) \qquad (26)$$

for the equations (14) and (22), and R is a constant. Given the initial condition as follows,

$$u_0 = R\sin(\phi)\cos(\theta) + R\sin(\phi)\sin(\theta) + R\cos(\phi)$$

Because we administer semi-implicit second-order Crank-Nicolson method in spatiotemporal term, we could choose the bigger Δt. It leads to spending less time to get our numerical solution. According to CFL condition, the stability of the Crank-Nicolson method in the equations (14) and (22) needs to satisfy the property

$$\Delta t = O(\Delta x)$$

In our programming, we could pick out

$$\Delta t = \frac{1}{2\pi}\Delta x = \frac{1}{N}.$$

It leads to faster computation. In next two Section 3.2.1 and Section 3.2.2, we derive the numerical solution at $T = 1$ and set $\Delta t = \frac{1}{N}$. Here provide the exact solution at $T = 1$ for (14) and (22) as follows,

$$u = e^{-2\frac{1}{R^2}}(R\sin(\phi)\cos(\theta) + R\sin(\phi)\sin(\theta) + R\cos(\phi))$$

3.2.1 Results of Solvers with the Central Difference Method

We solved the heat equation (14) $u_t = \Delta_s u$ on a spherical surface domain by the numerical methods that are FFT, second-order central difference method, Crank-Nicolson method and Thomas algorithm. In Table 1, solve (14) numerically with $R = 1$ at $T = 1$ and check them by the exact solution

$$u = e^{-2}(\sin(\phi)\cos(\theta) + \sin(\phi)\sin(\theta) + \cos(\phi)) \tag{27}$$

$\mathbf{M \times N}$	L_{∞}nrom	RATIO	ORDER
8×16	0.0058	0	0
16×32	0.0014	4.1429	2.0506
32×64	3.5098e-004	3.9888	1.9960
64×128	8.7537e-005	4.0095	2.0034
128×256	2.1885e-005	3.9999	2.0000

Table 1: The numerical results are second-order accuracy.

In Table 2, solve (14) numerically with $R = 5$ at $T = 1$ and check them by the exact solution

$$u = e^{-\frac{2}{25}}(5\sin(\phi)\cos(\theta) + 5\sin(\phi)\sin(\theta) + 5\cos(\phi)) \tag{28}$$

$\mathbf{M \times N}$	L_{∞}nrom	RATIO	ORDER
8×16	0.0050	0	0
16×32	0.0011	4.5455	2.1844
32×64	2.7313e-004	4.0274	2.0098
64×128	6.8221e-005	4.0036	2.0013
128×256	1.7053e-005	4.0005	2.0002

Table 2: The numerical results are second-order accuracy.

3.2.2 Results of Solvers with the Symmetric Discretization

We solved the heat equation (22) $u_t = \Delta_s u$ on a spherical surface domain by numerical methods that are FFT, symmetric discretization, Crank-Nicolson method, and Thomas algorithm.

In Table 3, solve (22) numerically with $R = 1$ at $T = 1$ and check them by the exact solution (27).

$M \times N$	L_∞ nrom	RATIO	ORDER
8×16	0.0090	0	0
16×32	0.0022	4.0909	2.0324
32×64	5.4423e-004	4.0424	2.0152
64×128	1.3612e-004	3.9982	1.9993
128×256	3.4027e-005	4.0004	2.0001

Table 3: The numerical results are second-order accuracy.

In Table 4, solve (22) numerically with $R = 5$ at $T = 1$ and check them by the exact solution (28).

$\mathbf{M \times N}$	L_∞ nrom	RATIO	ORDER
8×16	0.0066	0	0
16×32	0.0015	4.4000	2.1375
32×64	3.7037e-004	4.0500	2.0179
64×128	9.2165e-005	4.0186	2.0067
128×256	2.2994e-005	4.0082	2.0030

Table 4: The numerical results are second-order accuracy.

3.3 Mass Conservation

We developed an interest in keeping a lookout for the variation of the total mass in our numerical solvers due to

$$\frac{d}{dt}\iiint_{\partial\Omega} u \, dxdydz = \frac{d}{dt}\int_0^{2\pi}\int_0^{\pi} u \left|\frac{\partial \mathbf{X}}{\partial \phi} \times \frac{\partial \mathbf{X}}{\partial \theta}\right| d\phi d\theta = 0 \tag{29}$$

$$\partial\Omega\text{: spherical surface}$$

$$\mathrm{x} = \mathrm{R}\sin\phi\cos\theta, \mathrm{y} = \mathrm{R}\sin\phi\sin\theta, \mathrm{z} = \mathrm{R}\cos\phi$$

$$\mathbf{X}(\mathrm{R}, \phi, \theta) = (\mathrm{x}\,, \mathrm{y}\,, \mathrm{z}) = (\mathrm{R}\sin\phi\cos\theta\,, \mathrm{R}\sin\phi\sin\theta\,, \mathrm{R}\cos\phi)$$

$$\mathbf{X}_{\mathrm{i,j}} \approx \mathbf{X}(\phi_{\mathrm{i}}, \theta_{\mathrm{j}})$$

$$\frac{\partial\mathbf{X}}{\partial\phi}\times\frac{\partial\mathbf{X}}{\partial\theta} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ \frac{\partial \mathrm{x}}{\partial\phi} & \frac{\partial \mathrm{y}}{\partial\phi} & \frac{\partial \mathrm{z}}{\partial\phi} \\ \frac{\partial \mathrm{x}}{\partial\theta} & \frac{\partial \mathrm{y}}{\partial\theta} & \frac{\partial \mathrm{z}}{\partial\theta} \end{vmatrix} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ \mathrm{R}\cos\phi\cos\theta & \mathrm{R}\cos\phi\sin\theta & -\mathrm{R}\sin\phi \\ -\mathrm{R}\sin\phi\sin\theta & \mathrm{R}\sin\phi\cos\theta & 0 \end{vmatrix}$$

$$= \left(\mathrm{R}^2\sin^2\phi\cos\theta, \mathrm{R}^2\sin^2\phi\sin\theta, \mathrm{R}^2\sin\phi\cos\phi(\cos^2\theta + \sin^2\theta)\right)$$

$$= (\mathrm{R}^2\sin^2\phi\cos\theta, \mathrm{R}^2\sin^2\phi\sin\theta, \mathrm{R}^2\sin\phi\cos\phi)$$

$$\left|\frac{\partial\mathbf{X}}{\partial\phi}\times\frac{\partial\mathbf{X}}{\partial\theta}\right| = (\mathrm{R}^4\sin^4\phi(\cos^2\theta + \sin^2\theta) + \mathrm{R}^4\sin^2\phi\cos^2\phi)^{\frac{1}{2}}$$

$$= (\mathrm{R}^4\sin^4\phi + \mathrm{R}^4\sin^2\phi\cos^2\phi)^{\frac{1}{2}} = \left(\mathrm{R}^4\sin^2\phi(\sin^2\phi + \cos^2\phi)\right)^{\frac{1}{2}}$$

$$= \mathrm{R}^2|\sin\phi|$$

Choose specific four cases in our paper to observe the total mass of them which changes with time. Because of immovability of our spherical surface domain, the total mass is equal to

$$\int_0^{2\pi}\int_0^{\pi} \mathrm{u}\left|\frac{\partial\mathbf{X}}{\partial\phi}\times\frac{\partial\mathbf{X}}{\partial\theta}\right| \mathrm{d}\phi\mathrm{d}\theta \approx \sum_{\substack{\mathrm{i}=1,\dots,\mathrm{M}\\ \mathrm{j}=1,\dots,\mathrm{N}}} \mathrm{u}_{\mathrm{i,j}}\left|\frac{\partial\mathbf{X}_{\mathrm{i,j}}}{\partial\phi}\times\frac{\partial\mathbf{X}_{\mathrm{i,j}}}{\partial\theta}\right|\Delta\phi\Delta\theta = \sum_{\substack{\mathrm{i}=1,\dots,\mathrm{M}\\ \mathrm{j}=1,\dots,\mathrm{N}}} \mathrm{u}_{\mathrm{i,j}}\mathrm{R}^2|\sin\phi_{\mathrm{i}}|\,\Delta\phi\Delta\theta$$

The total mass at n-th time step is set by

$$\mathrm{S}^{\mathrm{n}} \equiv \sum_{\substack{\mathrm{i}=1,\dots,\mathrm{M}\\ \mathrm{j}=1,\dots,\mathrm{N}}} \mathrm{u}_{\mathrm{i,j}}^{\mathrm{n}}\mathrm{R}^2|\sin\phi_{\mathrm{i}}|\,\Delta\phi\Delta\theta$$

The relative error at n-th time step in our numerical solver means

$$\frac{|\mathrm{S}^{\mathrm{n}} - \mathrm{S}^0|}{\mathrm{S}^0}$$

3.3.1 The Mass of Solvers with the Central Difference Method

Initial settings as Figure 7: $\mathrm{M} = 32, \mathrm{N} = 64,\ \mathrm{R} = 5, \Delta\mathrm{t} = \frac{1}{2\pi}\Delta\mathrm{x} = \frac{1}{\mathrm{N}}$.

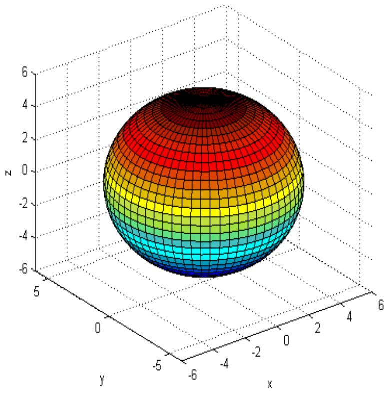


Figure 7: The domain is a spherical surface with $R = 5$.

Case 1: Pizza-like initial condition as follows,

```
u(1:0.5*M,0.5*N:0.5*N+5)=1; others are set to be zero.
```

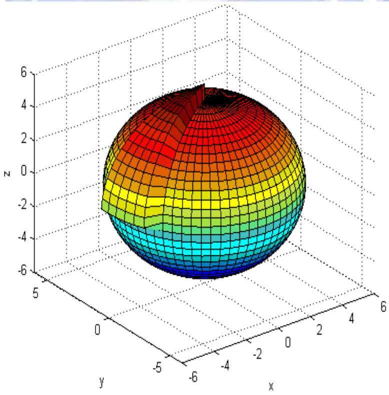


Figure 8: Initial value u; $T = 0$; $\text{Total mass} = 14.7321$.

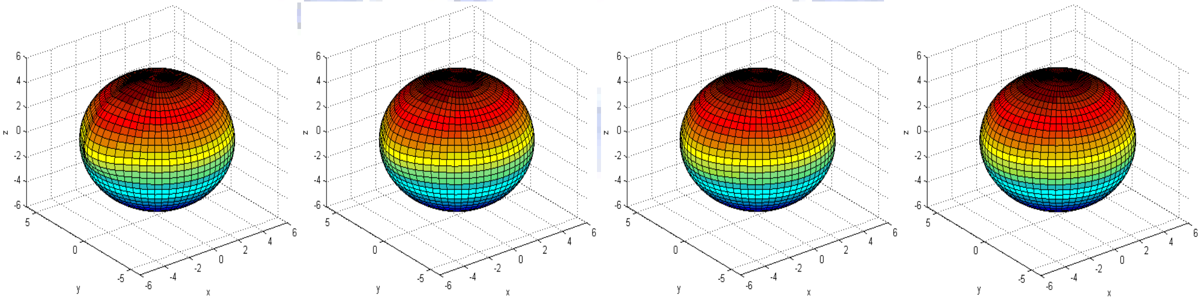


Figure 9: Left, $T = 0.25$, $\text{mass} = 14.7321$, $\text{relative error} = 1.8087\text{e} - 015$.
Mid-left, $T = 0.5$, $\text{mass} = 14.7321$, $\text{relative error} = 7.3552\text{e} - 015$.
Mid-right, $T = 0.75$, $\text{mass} = 14.7321$, $\text{relative error} = 9.8873\text{e} - 015$.
Right, $T = 0.1$, $\text{mass} = 14.7321$, $\text{relative error} = 1.2058\text{e} - 014$.

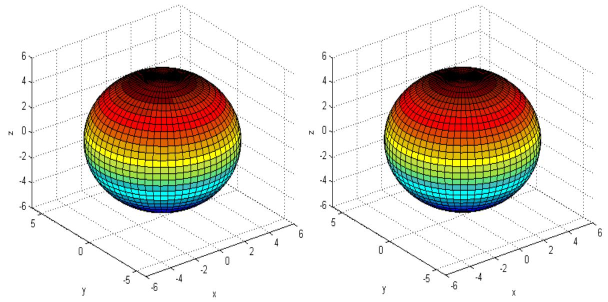


Figure 10: Left, $\mathrm{T} = 5$, mass $= 14.7321$, relative error $= 7.8099\mathrm{e} - 014$.
Right, $\mathrm{T} = 20$, mass $= 14.7321$, relative error $= 2.8661\mathrm{e} - 013$.

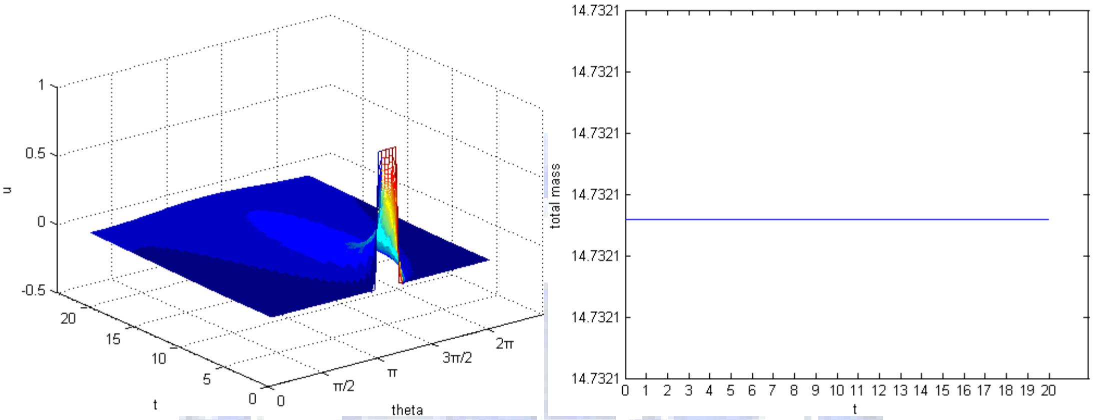


Figure 11: Left, the values of u on the equator change with time.
Right, total mass of u changes with time in $0 \sim 20$ seconds.

According to Figure 9, Figure 10, Figure 11, u was diffused as time goes by to be like the look of the original domain. Consider that total mass has little machine error. Therefore, Case 1 complies with the mass conservation law.

Case 2: Initial condition was set as one point as follows,

```
u(0.5*M,0.5*N+5)=1; others are set to be zero.
```

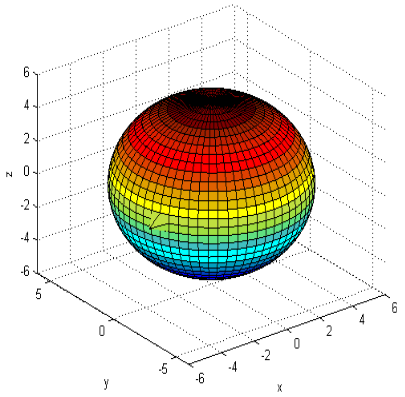


Figure 12: Initial value u; $\mathrm{T} = 0$; Total mass $= 0.24067$.

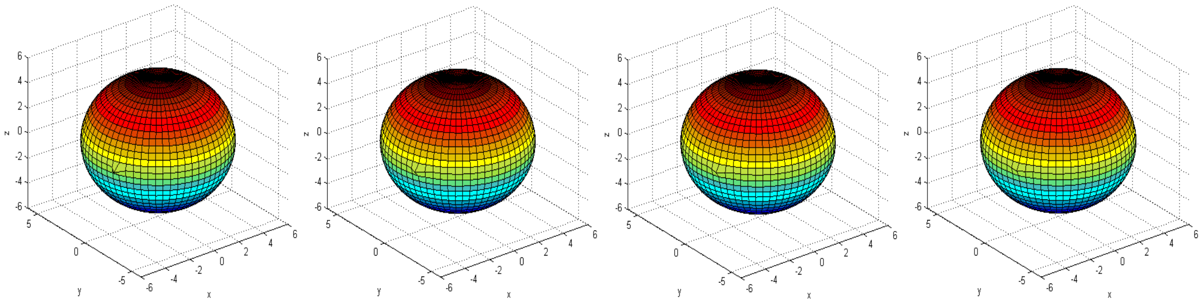


Figure 13：Left, $\mathrm{T} = 1/64$, mass $= 0.24067$, relative error $= 5.0167\mathrm{e} - 007$.

Mid-left, $\mathrm{T} = 1/32$, mass $= 0.24067$, relative error $= 1.0033\mathrm{e} - 006$.

Mid-right, $\mathrm{T} = 3/64$, mass $= 0.24067$, relative error $= 1.505\mathrm{e} - 006$.

Right, $\mathrm{T} = 1/16$, mass $= 0.24067$, relative error $= 2.0067\mathrm{e} - 006$.

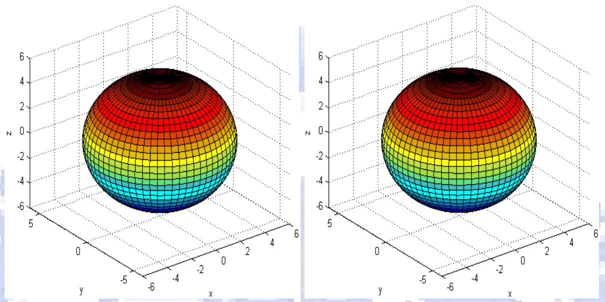


Figure 14：Left, $\mathrm{T} = 5$, mass $= 0.24063$, relative error $= 0.00014728$.

Right, $\mathrm{T} = 20$, mass $= 0.24061$, relative error $= 0.00024238$.

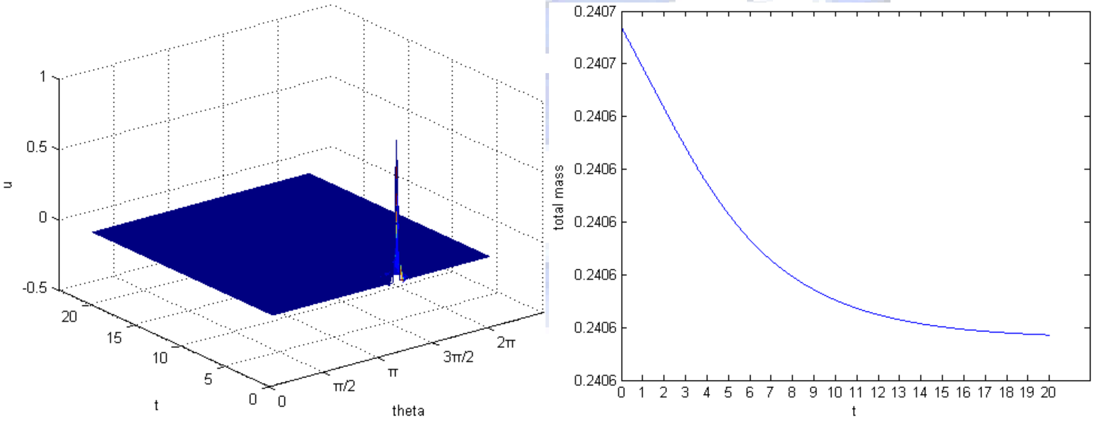


Figure 15：Left, the values of u on the equator change with time.

Right, total mass of u changes with time in $0{\sim}20$ seconds.

According to Figure 13, Figure 14, Figure 15, u was diffused as time goes by to be like the look of the original domain. Total mass is getting decreasing a little bit, but it will turn to stable that means mass will not change any more in a long time. Case 2 does not comply with the mass conservation law.

Case 3: Chapeau-like initial condition as follows,

```
u(1:3,1:N)=1; others are set to be zero.
```

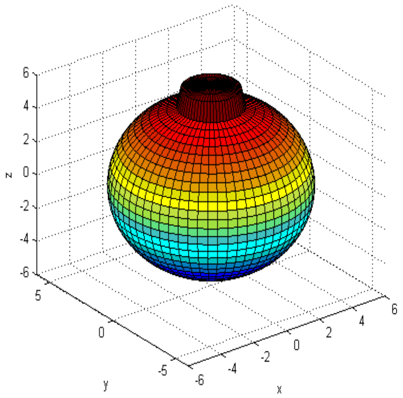


Figure 16: Initial value u; $T = 0$; Total mass $= 6.7665$.

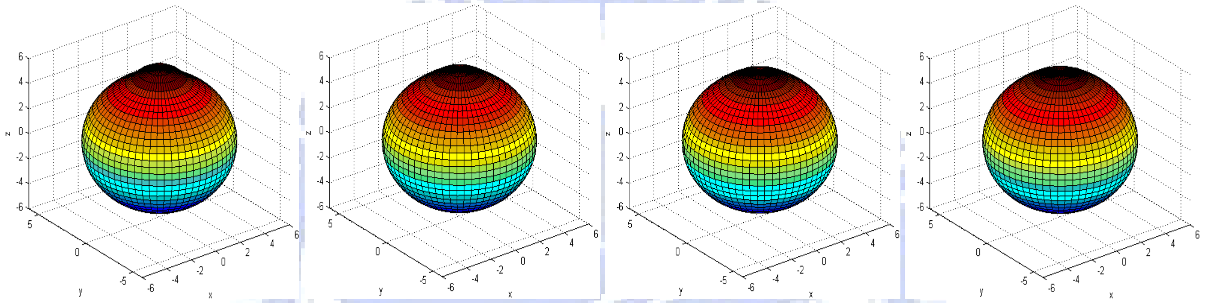


Figure 17: Left, $T = 0.25$, mass $= 6.7665$, relative error $= 0.00017029$.
Mid-left, $T = 0.5$, mass $= 6.7686$, relative error $= 0.00030516$.
Mid-right, $T = 0.75$, mass $= 6.7693$, relative error $= 0.00040749$.
Right, $T = 1$, mass $= 6.7698$, relative error $= 0.00040878$.

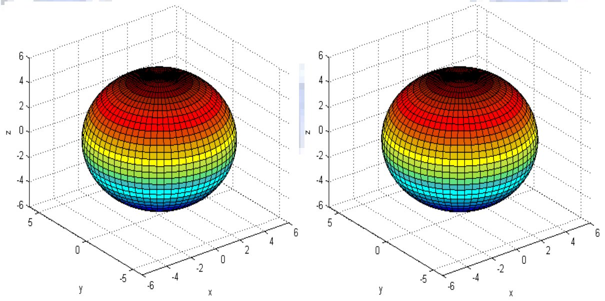


Figure 18: Left, $T = 5$, mass $= 6.773$, relative error $= 0.00095451$.
Right, $T = 20$, mass $= 6.7743$, relative error $= 0.001145$.

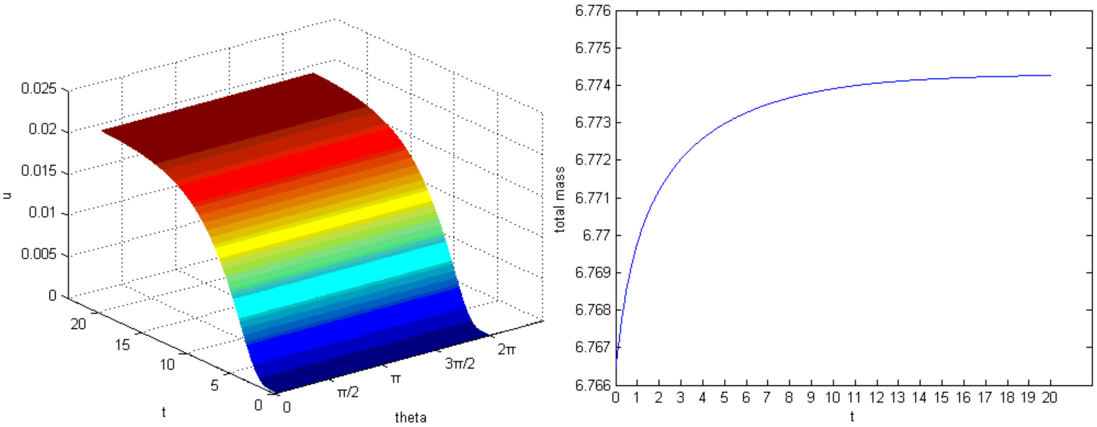


Figure 19: Left, the values of u on the equator change with time.
Right, total mass of u changes with time in 0~20 seconds.

According to Figure 17, Figure 18, Figure 19, u was diffused as time goes by to be like the look of the original domain. Total mass is getting increasing a little bit, but it will turn to stable that means mass will not change any more in a long time. Case 3 does not comply with the mass conservation law.

Case 4: Smooth initial condition as follows,

$$u(m,n)=abs(sin((m-0.5)*\frac{\pi}{M}))+abs(cos((n-1)*\frac{2\pi}{N})),\ \ \forall\, m, n.$$

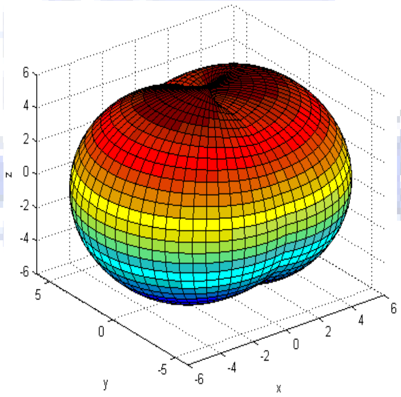


Figure 20: Initial value u; T = 0; Total mass = 446.6597.

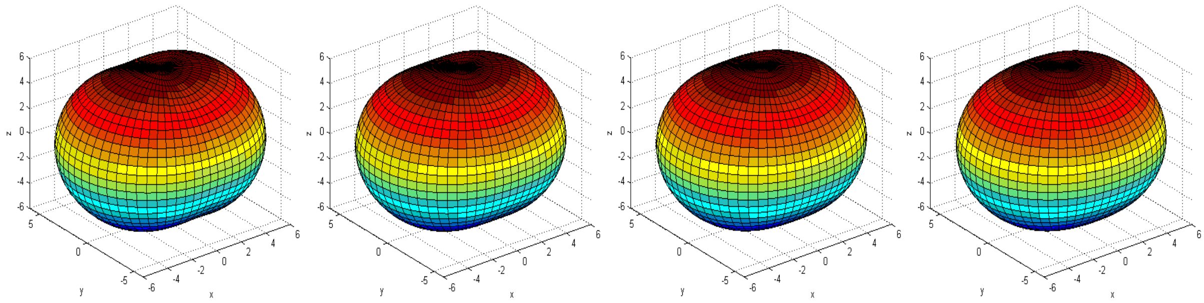


Figure 21: Left, T = 0.25, mass = 446.658, relative error = 3.7503e − 006.

Mid-left, $T = 0.5$, mass $= 446.6566$, relative error $= 6.9874e-006$.

Mid-right, $T = 0.75$, mass $= 446.6553$, relative error $= 9.8891e-006$.

Right, $T = 1$, mass $= 446.6541$, relative error $= 1.2529e-006$.

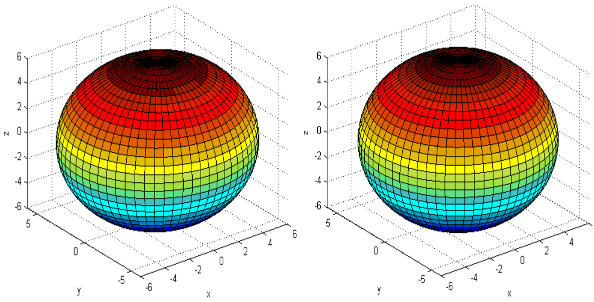


Figure 22: Left, $T = 5$, mass $= 446.6434$, relative error $= 3.6478e-005$.

Right, $T = 20$, mass $= 446.6373$, relative error $= 5.0161e-005$.

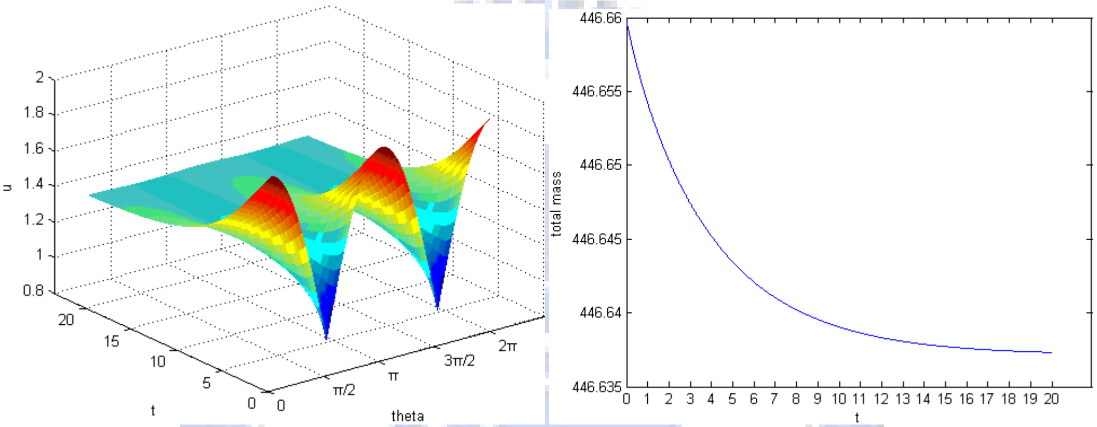


Figure 23: Left, the values of u on the equator change with time.

Right, total mass of u changes with time in $0 \sim 20$ seconds.

According to Figure 21, Figure 22, Figure 23, u was diffused as time goes by to be like the look of the original domain. Total mass is getting decreasing a little bit, but it will turn to stable that means mass will not change any more in a long time. Case 4 does not comply with the mass conservation law.

In above Case 1~Case 4, we derive that Case 1 can get the perfect result by mass conservation with some machine error；the others show up a little bit derivation that is less than one percent in our relative error.

3.3.2 The Mass of Solvers with the Symmetric Discretization

Initial settings as Figure 7: $M = 32, N = 64,\ R = 5, \Delta t = \frac{1}{2\pi}\Delta x = \frac{1}{N}$.

Case 1: Initial values are the same as Figure 8,

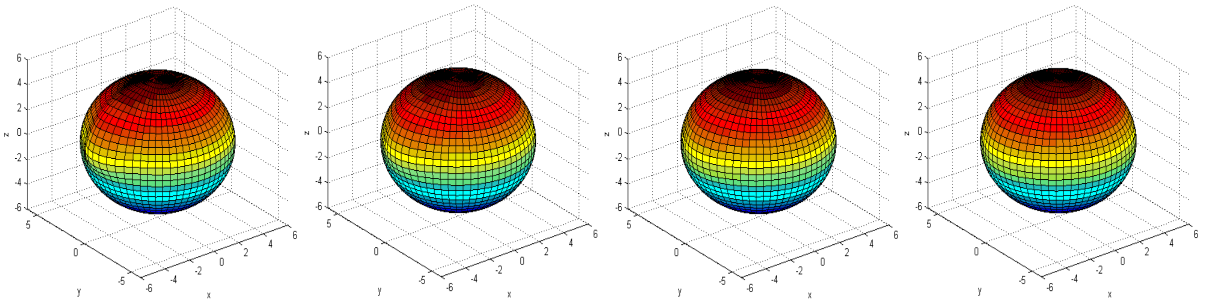


Figure 24: Left, $\mathrm{T} = 0.25$, $\text{mass} = 14.7321$, $\text{relative error} = 1.2058\mathrm{e} - 015$.
Mid-left, $\mathrm{T} = 0.5$, $\text{mass} = 14.7321$, $\text{relative error} = 4.8231\mathrm{e} - 015$.
Mid-right, $\mathrm{T} = 0.75$, $\text{mass} = 14.7321$, $\text{relative error} = 1.4469\mathrm{e} - 015$.
Right, $\mathrm{T} = 0.1$, $\text{mass} = 14.7321$, $\text{relative error} = 1.6881\mathrm{e} - 015$.

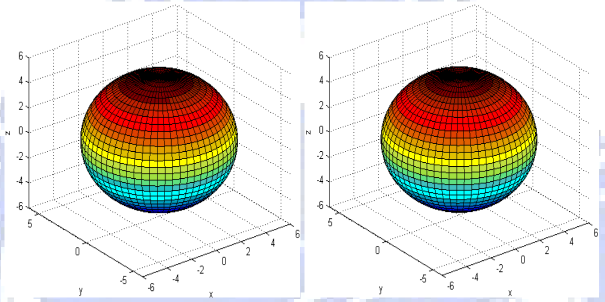


Figure 25: Left, $\mathrm{T} = 5$, $\text{mass} = 14.7321$, $\text{relative error} = 5.6671\mathrm{e} - 015$.
Right, $\mathrm{T} = 20$, $\text{mass} = 14.7321$, $\text{relative error} = 1.1455\mathrm{e} - 014$.

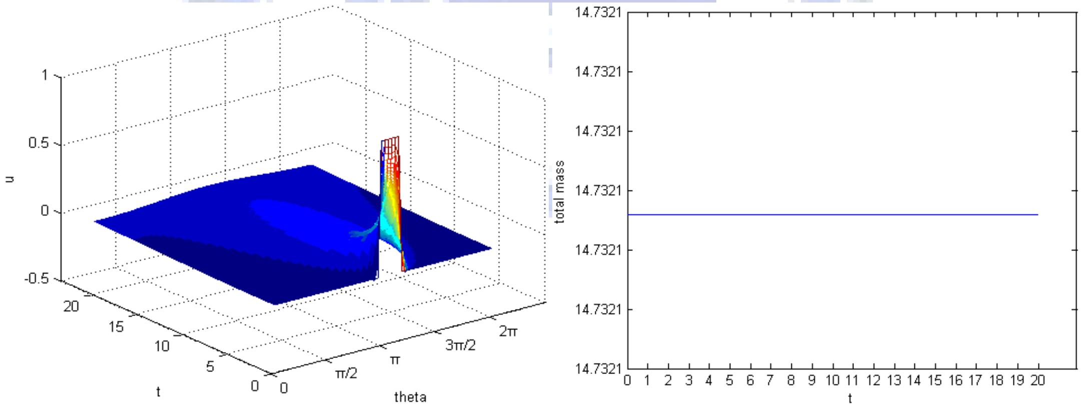


Figure 26: Left, the values of u on the equator change with time.
Right, total mass of u changes with time in 0~20 seconds.

According to Figure 24, Figure 25, Figure 26, u was diffused as time goes by to be like the look of the original domain. Consider that total mass has little machine error. Therefore, Case 1 complies with the mass conservation law.

Case 2: Initial values are the same as Figure 12,

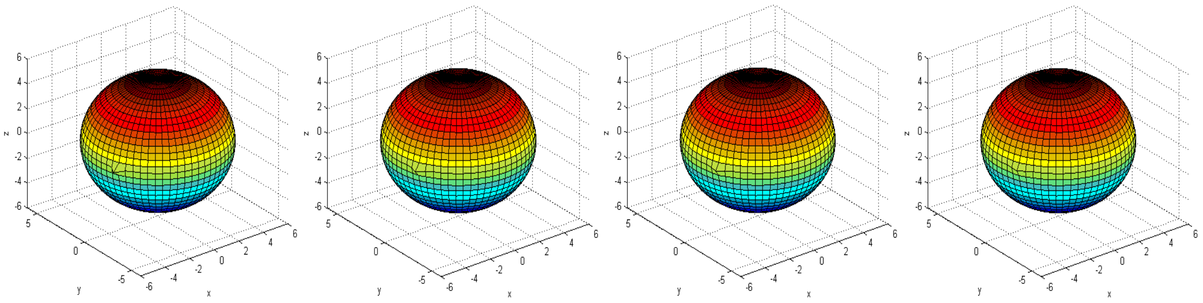


Figure 27： Left, $\mathrm{T} = 1/64$, mass $= 0.24067$, relative error $= 6.9197\mathrm{e} - 016$.

Mid-left, $\mathrm{T} = 1/32$, mass $= 0.24067$, relative error $= 8.0729\mathrm{e} - 016$..

Mid-right, $\mathrm{T} = 3/64$, mass $= 0.24067$, relative error $= 1.1533\mathrm{e} - 016$.

Right, $\mathrm{T} = 1/16$, mass $= 0.24067$, relative error $= 2.3066\mathrm{e} - 015$.

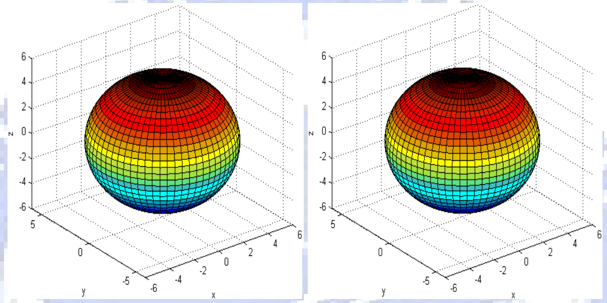


Figure 28： Left, $\mathrm{T} = 5$, mass $= 0.24067$, relative error $= 5.6511\mathrm{e} - 015$.

Right, $\mathrm{T} = 20$, mass $= 0.24067$, relative error $= 4.9591\mathrm{e} - 015$.

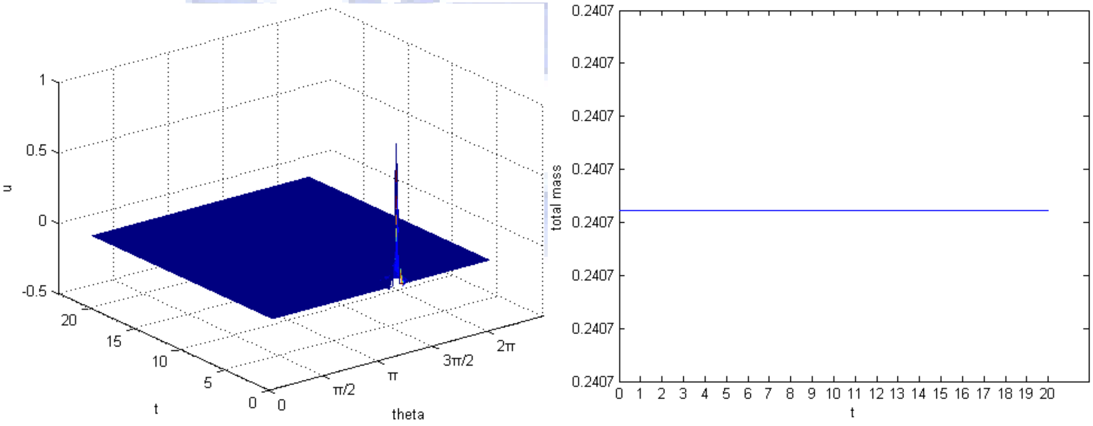


Figure 29： Left, the values of u on the equator change with time.

Right, total mass of u changes with time in $0{\sim}20$ seconds.

According to Figure 27, Figure 28, Figure 29, u was diffused as time goes by to be like the look of the original domain. Consider that total mass has little machine error. Therefore, Case 2 complies with the mass conservation law.

Case 3: Initial values are the same as Figure 16,

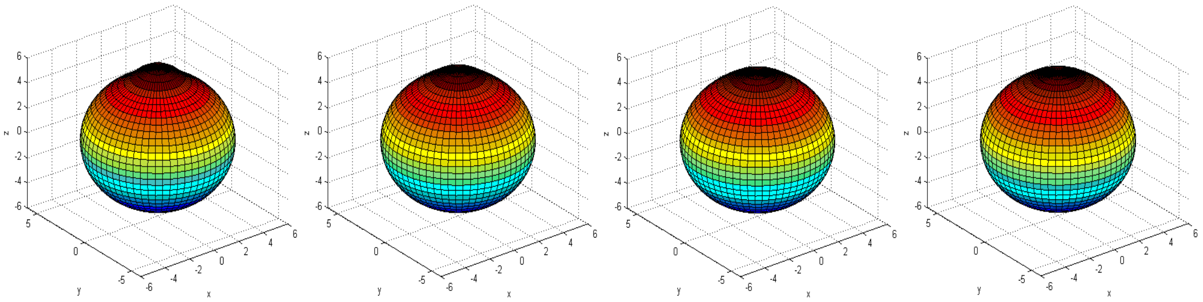


Figure 30: Left, $\mathrm{T} = 0.25$, $\mathrm{mass} = 6.7665$, $\mathrm{relative\ error} = 1.3126\mathrm{e} - 016$.
Mid-left, $\mathrm{T} = 0.5$, $\mathrm{mass} = 6.7665$, $\mathrm{relative\ error} = 3.019\mathrm{e} - 016$..
Mid-right, $\mathrm{T} = 0.75$, $\mathrm{mass} = 6.7665$, $\mathrm{relative\ error} = 0$.
Right, $\mathrm{T} = 1$, $\mathrm{mass} = 6.7665$, $\mathrm{relative\ error} = 4.8567\mathrm{e} - 016$.

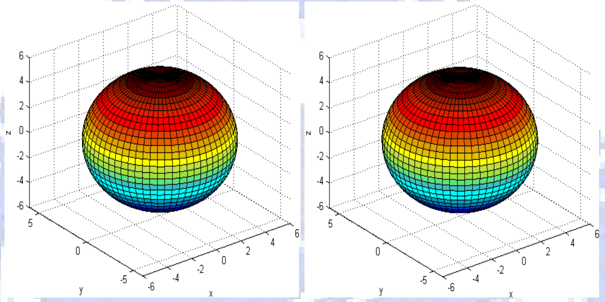


Figure 31: Left, $\mathrm{T} = 5$, $\mathrm{mass} = 6.7665$, $\mathrm{relative\ error} = 1.3126\mathrm{e} - 015$.
Right, $\mathrm{T} = 20$, $\mathrm{mass} = 6.7665$, $\mathrm{relative\ error} = 6.563\mathrm{e} - 016$.

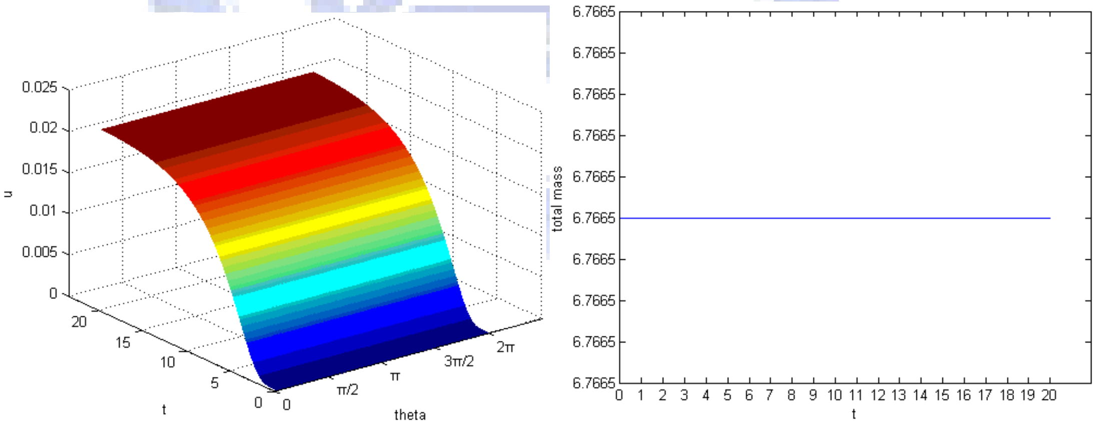


Figure 32: Left, the values of u on the equator change with time.
Right, total mass of u changes with time in $0{\sim}20$ seconds.

According to Figure 30, Figure 31, Figure 32, u was diffused as time goes by to be like the look of the original domain. Consider that total mass has little machine error. Therefore, Case 3 complies with the mass conservation law.

Case 4: Initial values are the same as Figure 20,

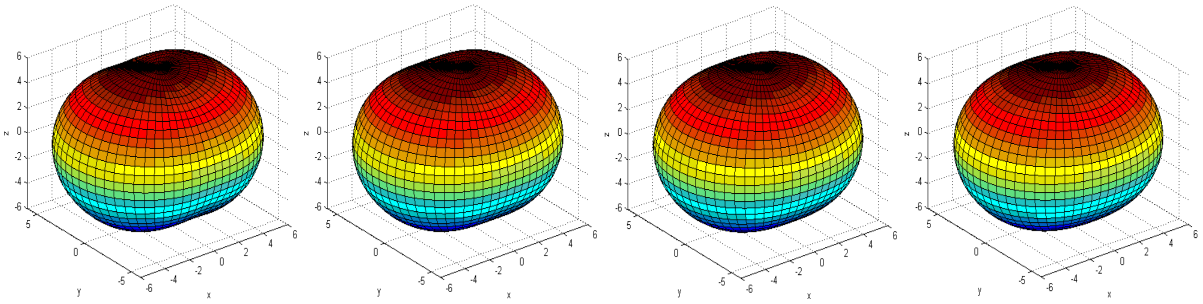


Figure 33: Left, $\mathrm{T} = 0.25$, $\mathrm{mass} = 446.6597$, $\mathrm{relative\,error} = 6.1086\mathrm{e} - 015$.
Mid-left, $\mathrm{T} = 0.5$, $\mathrm{mass} = 446.6597$, $\mathrm{relative\,error} = 3.8179\mathrm{e} - 016$.
Mid-right, $\mathrm{T} = 0.75$, $\mathrm{mass} = 446.6597$, $\mathrm{relative\,error} = 3.4361\mathrm{e} - 015$
Right, $\mathrm{T} = 1$, $\mathrm{mass} = 446.6597$, $\mathrm{relative\,error} = 3.0543\mathrm{e} - 015$.

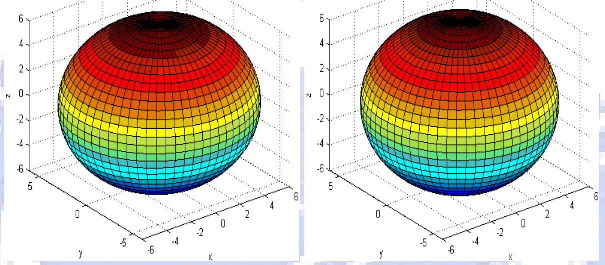


Figure 34: Left, $\mathrm{T} = 5$, $\mathrm{mass} = 446.6597$, $\mathrm{relative\,error} = 4.1997\mathrm{e} - 015$.
Right, $\mathrm{T} = 20$, $\mathrm{mass} = 446.6597$, $\mathrm{relative\,error} = 3.3088\mathrm{e} - 015$.

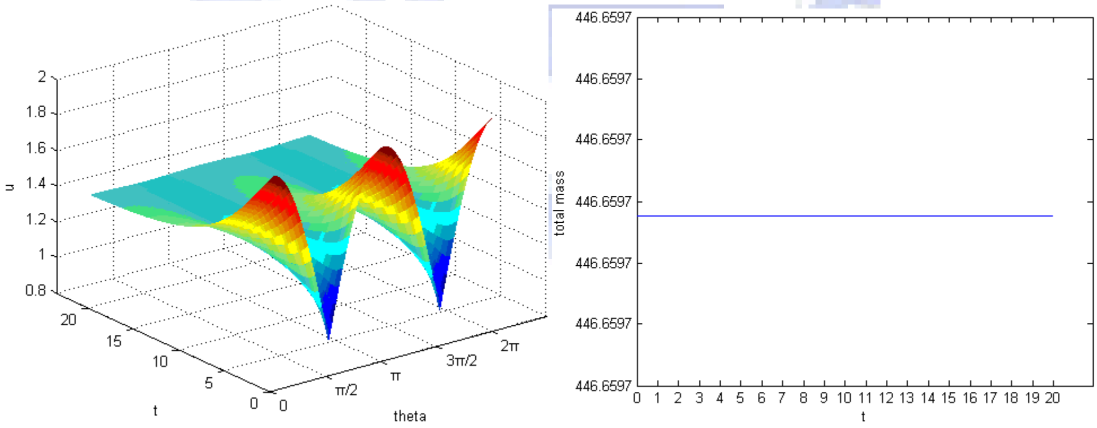


Figure 35: Left, the values of u on the equator change with time.
Right, total mass of u changes with time in $0 \sim 20$ seconds.

According to Figure 33, Figure 34, Figure 35, u was diffused as time goes by to be like the look of the original domain. Consider that total mass has little machine error. Therefore, Case 4 complies with the mass conservation law.

In above Case 1~Case 4, we derive that all these cases can comply with the mass conservation law and leave some machine error here.

3.3.3 Comparison between the Central Difference Method and the Symmetric Discretization in Spherical Coordinates

According to the results in Section 3.3.1 and Section 3.3.2, we could summarize the following table:

	3.3.1	**3.3.2**
Case1	2.8661e-013	1.1455e-014
Case2	0.00024238	4.9591e-015
Case3	0.001145	6.563e-016
Case4	5.0161e-005	3.3088e-015

Table 5: The relative error in four cases with different numerical methods at $T = 20$.

We are satisfied with the results in the four cases of the solvers in Section 3.3.2 because they comply with the mass conservation law and they only leave indelible and negligible machine error with us. Before discussing farther, introduce that why these four cases are handpicked as follows:

Case 1: The initial values are given from the pole to the equator, and they cross through the dense grids near the poles and the dissipative grids near the equator. This case contains all situations which could average the diversity of geometry.

Case 2: The initial value is given one point. The case could extend to all cases which are composed of some discrete points.

Case 3: The initial value capped the poles which are the densest position in our grids. We could understand how the values in the poles which are usually the singularities in.

Case 4: We give the smooth initial values which are different from preceding three cases.

Due to the results of above illustrations, we believe that the solvers in Section 3.3.2 will make any initial condition keep the mass conservation law. Table 5 brings us that the symmetric discretization is better than central difference method in the mass. By the way that Case 1 complies with the mass conservation law both in Section 3.3.1 and Section 3.3.2.

4、Fast Heat Solver in Ellipsoid Coordinates

4.1 Heat Solver on an Ellipsoid Surface Domain

4.1.1 Solvers with the Central Difference Method

According to [11], we have the formula of the heat equation on an ellipsoid surface domain in ellipsoid coordinates as

$$\begin{cases} u_t = \Delta_s u \\ \Delta_s u = \nabla_s^2 u = \dfrac{1}{h^2}\left(\dfrac{\partial^2 u}{\partial \phi^2} + \cot(\phi)\dfrac{\partial u}{\partial \phi} + \left(\dfrac{1}{\sinh^2(\beta)} + \dfrac{1}{\sin^2(\phi)}\right)\dfrac{\partial^2 u}{\partial \theta^2}\right) \end{cases} \tag{30}$$

Here

$$h^2 = \alpha^2(\sinh^2(\beta) + \sin^2(\phi))$$

$$x = \alpha\sinh\beta\sin\phi\cos\theta\ , x = \alpha\sinh\beta\sin\phi\sin\theta, z = \alpha\cosh\beta\cos\phi$$

In (30), ignore α and β by setting they be constants, and denote

$$u(\alpha, \beta, \phi, \theta, t) \approx u(\phi, \theta, t)$$

The setting of grids is the same as Section 3.1.2.

Step 1: Similar to Section 3.1.2 Step 1.

Step 2: Similar to Section 3.1.2 Step 2 and substitute (18) into (30),

$$\Delta_s u = \frac{1}{h^2}\left(\frac{\partial^2 \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} u_k(\phi)\, e^{ik\theta}}{\partial \phi^2} + \cot(\phi)\frac{\partial \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} u_k(\phi)\, e^{ik\theta}}{\partial \phi} + \left(\frac{1}{\sinh^2(\beta)} + \frac{1}{\sin^2(\phi)}\right)\frac{\partial^2 \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} u_k(\phi)\, e^{ik\theta}}{\partial \theta^2}\right)$$

$$= \frac{1}{h^2}\left(\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \frac{\partial^2 u_k(\phi) e^{ik\theta}}{\partial \phi^2} + \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \cot(\phi)\frac{\partial u_k(\phi) e^{ik\theta}}{\partial \phi} + \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \left(\frac{1}{\sinh^2(\beta)} + \frac{1}{\sin^2(\phi)}\right)\frac{\partial^2 u_k(\phi) e^{ik\theta}}{\partial \theta^2}\right)$$

$$= \frac{1}{h^2}\left(\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \frac{\partial^2 u_k(\phi)}{\partial \phi^2} e^{ik\theta} + \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \cot(\phi)\frac{\partial u_k(\phi)}{\partial \phi} e^{ik\theta} + \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \left(\frac{1}{\sinh^2(\beta)} + \frac{1}{\sin^2(\phi)}\right) u_k(\phi)(-k^2)\, e^{ik\theta}\right)$$

$$= \frac{1}{h^2}\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\left(\frac{\partial^2 u_k(\phi)}{\partial\phi^2} + \cot(\phi)\frac{\partial u_k(\phi)}{\partial\phi} - k^2(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)})u_k(\phi)\right)e^{ik\theta} \tag{31}$$

Substitute the surface Laplacian term of (30) into (17),

$$\frac{1}{h^2}\left(\frac{\partial^2 u^{n+1}}{\partial\phi^2} + \cot(\phi)\frac{\partial u^{n+1}}{\partial\phi} + (\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)})\frac{\partial^2 u^{n+1}}{\partial\theta^2}\right) - \frac{2}{\Delta t}u^{n+1}$$
$$= -\frac{1}{h^2}(\frac{\partial^2 u^{n}}{\partial\phi^2} + \cot(\phi)\frac{\partial u^{n}}{\partial\phi} + (\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)})\frac{\partial^2 u^{n}}{\partial\theta^2}) - \frac{2}{\Delta t}u^{n} \tag{32}$$

Substitute (18) into (32), and from (31) we can derive

$$\frac{1}{h^2}\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\left(\frac{\partial^2 u_k^{n+1}(\phi)}{\partial\phi^2} + \cot(\phi)\frac{\partial u_k^{n+1}(\phi)}{\partial\phi} - k^2(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)})u_k^{n+1}(\phi)\right)e^{ik\theta} - \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\frac{2}{\Delta t}u_k^{n+1}(\phi)\,e^{ik\theta}$$

$$= -\frac{1}{h^2}\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}(\frac{\partial^2 u_k^{n}(\phi)}{\partial\phi^2} + \cot(\phi)\frac{\partial u_k^{n}(\phi)}{\partial\phi} - k^2(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)})u_k^{n}(\phi))\,e^{ik\theta} - \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\frac{2}{\Delta t}u_k^{n}(\phi)\,e^{ik\theta}$$

$$=> \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\left(\frac{1}{h^2}\left(\frac{\partial^2 u_k^{n+1}(\phi)}{\partial\phi^2} + \cot(\phi)\frac{\partial u_k^{n+1}(\phi)}{\partial\phi} - k^2(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)})u_k^{n+1}(\phi)\right) - \frac{2}{\Delta t}u_k^{n+1}(\phi)\right)e^{ik\theta}$$

$$= \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\left(-\frac{1}{h^2}\left(\frac{\partial^2 u_k^{n}(\phi)}{\partial\phi^2} + \cot(\phi)\frac{\partial u_k^{n}(\phi)}{\partial\phi} - k^2(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)})u_k^{n}(\phi)\right) - \frac{2}{\Delta t}u_k^{n}(\phi)\right)e^{ik\theta}$$

Eliminate $\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}$ and $e^{ik\theta}$ from both sides of the equation,

$$\frac{1}{h^2}\left(\frac{\partial^2 u_k^{n+1}(\phi)}{\partial\phi^2} + \cot(\phi)\frac{\partial u_k^{n+1}(\phi)}{\partial\phi} - k^2(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)})u_k^{n+1}(\phi)\right) - \frac{2}{\Delta t}u_k^{n+1}(\phi)$$

$$= -\frac{1}{h^2}\left(\frac{\partial^2 u_k^{n}(\phi)}{\partial\phi^2} + \cot(\phi)\frac{\partial u_k^{n}(\phi)}{\partial\phi} - k^2(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)})u_k^{n}(\phi)\right) - \frac{2}{\Delta t}u_k^{n}(\phi)$$

After multiplying by h^2 to both sides of the equation and equaling the Fourier coefficients, $u_k(\phi)$ satisfies

$$\frac{\partial^2 u_k^{n+1}(\phi)}{\partial\phi^2} + \cot(\phi)\frac{\partial u_k^{n+1}(\phi)}{\partial\phi} - k^2(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)})u_k^{n+1}(\phi) - \frac{2h^2}{\Delta t}u_k^{n+1}(\phi)$$
$$= -\frac{\partial^2 u_k^{n}(\phi)}{\partial\phi^2} - \cot(\phi)\frac{\partial u_k^{n}(\phi)}{\partial\phi} + k^2\left(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)}\right)u_k^{n}(\phi) - \frac{2h^2}{\Delta t}u_k^{n}(\phi)$$

The Step 2 is equal to do FFT for θ as Figure 4.

Step 3: We use the central difference method for ϕ, i.e.

$$\frac{\partial U_i}{\partial \phi} \approx \frac{U_{i+1} - U_{i-1}}{2\Delta\phi} \quad \text{and} \quad \frac{\partial^2 U_i}{\partial \phi^2} \approx \frac{U_{i+1} - 2U_i + U_{i-1}}{\Delta\phi^2}$$

For each Fourier modes k ($k = -\frac{N}{2}, -\frac{N}{2} + 1, \ldots, 0, 1, \ldots, \frac{N}{2} - 1.$), we set

$$u_k^{n+1}(\phi) = U^{n+1} \text{ at } T = (n+1)\Delta t$$

$$u_k^{n}(\phi) = U^{n} \text{ at } T = n\Delta t$$

and solve the following equations.

$$\frac{U_{i+1}^{n+1} - 2U_i^{n+1} + U_{i-1}^{n+1}}{\Delta\phi^2} + \cot(\phi_i)\frac{U_{i+1}^{n+1} - U_{i-1}^{n+1}}{2\Delta\phi} - k^2\left(\frac{1}{\sinh^2(\beta)} + \frac{1}{\sin^2(\phi_i)}\right)U_i^{n+1} - \frac{2h^2}{\Delta t}U_i^{n+1}$$
$$= -\frac{U_{i+1}^{n} - 2U_i^{n} + U_{i-1}^{n}}{\Delta\phi^2} - \cot(\phi_i)\frac{U_{i+1}^{n} - U_{i-1}^{n}}{2\Delta\phi} + k^2\left(\frac{1}{\sinh^2(\beta)} + \frac{1}{\sin^2(\phi_i)}\right)U_i^{n} - \frac{2h^2}{\Delta t}U_i^{n}$$

for $i = 1, \ldots, M$.

Both sides of the equations were multiplied by $\Delta\phi^2$ and rearrange them,

$$\left(1 - \frac{\cot(\phi_i)\Delta\phi}{2}\right)U_{i-1}^{n+1} + \left(-2 - k^2\Delta\phi^2\left(\frac{1}{\sinh^2(\beta)} + \frac{1}{\sin^2(\phi_i)}\right) - \frac{2h^2\Delta\phi^2}{\Delta t}\right)U_i^{n+1} + \left(1 + \frac{\cot(\phi_i)\Delta\phi}{2}\right)U_{i+1}^{n+1}$$
$$= -\left(1 - \frac{\cot(\phi_i)\Delta\phi}{2}\right)U_{i-1}^{n} + \left(2 + k^2\Delta\phi^2\left(\frac{1}{\sinh^2(\beta)} + \frac{1}{\sin^2(\phi_i)}\right) - \frac{2h^2\Delta\phi^2}{\Delta t}\right)U_i^{n} - \left(1 + \frac{\cot(\phi_i)\Delta\phi}{2}\right)U_{i+1}^{n} \quad (33)$$

Therefore, we could set that

$$A_i = 1 - \frac{\cot(\phi_i)\Delta\phi}{2}$$

$$B_i = -2 - k^2\Delta\phi^2\left(\frac{1}{\sinh^2(\beta)} + \frac{1}{\sin^2(\phi_i)}\right) - \frac{2h^2\Delta\phi^2}{\Delta t}$$

$$C_i = 1 + \frac{\cot(\phi_i)\Delta\phi}{2}$$

$$D_i = -B_i - \frac{4h^2\Delta\phi^2}{\Delta t}$$

for $i = 1, \ldots, M$.

Before solving the governing equation in ellipsoid coordinates as (30), we burned with curiosity over that the Fourier coefficients of a function in the coordinates satisfy the symmetry constrain [13]. Here we ignore the discussion by applying the similar arguments in Section 3.1.2 and derive,

$$U_0 \approx u_k(\phi_0) = u_k\left(-\frac{\Delta\phi}{2}\right) = (-1)^k u_k\left(\frac{\Delta\phi}{2}\right) \approx (-1)^k U_1$$

$$\mathrm{U}_{M+1} \approx \mathrm{u}_k(\phi_{M+1}) = \mathrm{u}_k\left(\pi + \frac{\Delta\phi}{2}\right) = (-1)^k \mathrm{u}_k\left(\pi - \frac{\Delta\phi}{2}\right) \approx (-1)^k \mathrm{U}_M$$

for each k-th Fourier coefficient. Then we could produce the following matrices,

$$\begin{bmatrix} (-1)^k A_1 + B_1 & C_1 & & & & \\ A_2 & B_2 & C_2 & & & \\ & . & . & . & & \\ & & A_{M-1} & B_{M-1} & & C_{M-1} \\ & & & A_M & & B_M + (-1)^k C_M \end{bmatrix} \begin{bmatrix} U_1^{n+1} \\ U_2^{n+1} \\ . \\ U_{M-1}^{n+1} \\ U_M^{n+1} \end{bmatrix}$$

$$= \begin{bmatrix} (-1)^{k+1} A_1 + D_1 & -C_1 & & & & \\ -A_2 & D_2 & -C_2 & & & \\ & . & . & . & & \\ & & -A_{M-1} & D_{M-1} & & -C_{M-1} \\ & & & -A_M & & D_M + (-1)^{k+1} C_M \end{bmatrix} \begin{bmatrix} U_1^{n} \\ U_2^{n} \\ . \\ U_{M-1}^{n} \\ U_M^{n} \end{bmatrix}$$

and must solve these linear systems N times.

The Step 3 is equal to solve the unknowns in the linear systems in each Fourier mode as Figure 5.

Step 4: Similar to Section 3.1.2 Step 4.

According to the procedure, i.e. Step 1=>Step 2=>Step 3=>Step 4, we could complete the numerical solution at one time step and we can derive the numerical solution at any time by keeping performing the procedure recursively.

4.1.2 Solvers with the Symmetric Discretization

According to [11], we have the formula for the heat equation on an ellipsoid surface domain in ellipsoid coordinates as

$$\begin{cases} \mathrm{u}_t = \Delta_s \mathrm{u} \\ \Delta_s \mathrm{u} = \nabla_s^2 \mathrm{u} = \dfrac{1}{\mathrm{h}^2}\left(\dfrac{1}{\sin(\phi)}\dfrac{\partial}{\partial\phi}(\sin(\phi)\dfrac{\partial \mathrm{u}}{\partial\phi}) + (\dfrac{1}{\sinh^2(\beta)} + \dfrac{1}{\sin^2(\phi)})\dfrac{\partial^2 \mathrm{u}}{\partial\theta^2}\right) \end{cases} \qquad (34)$$

The initial settings are the same as Section 4.1.1.

Step 1: Similar to Section 3.1.2 Step 1.

Step 2: Similar to Section 3.1.2 Step 2 and substitute (18) into (34),

$$\Delta_s u=\frac{1}{h^2}\left(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}\left(\sin\phi\frac{\partial\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}u_k(\phi)\,e^{ik\theta}}{\partial\phi}\right)+(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)})\frac{\partial^2\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}u_k(\phi)\,e^{ik\theta}}{\partial\theta^2}\right)$$

$$=\frac{1}{h^2}\left(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}(\sin\phi\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\frac{\partial u_k(\phi)e^{ik\theta}}{\partial\phi})+(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)})\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\frac{\partial^2 u_k(\phi)e^{ik\theta}}{\partial\theta^2}\right)$$

$$=\frac{1}{h^2}\left(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}(\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\sin\phi\frac{\partial u_k(\phi)}{\partial\phi}e^{ik\theta})+(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)})\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}u_k(\phi)(-k^2)e^{ik\theta}\right)$$

$$=\frac{1}{h^2}\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\left(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}(\sin\phi\frac{\partial u_k(\phi)}{\partial\phi})-k^2(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)})u_k(\phi)\right)e^{ik\theta}\qquad(35)$$

Substitute the surface Laplacian term of (34) into (17),

$$\frac{1}{h^2}\left(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}(\sin\phi\frac{\partial u^{n+1}}{\partial\phi})+(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)})\frac{\partial^2 u^{n+1}}{\partial\theta^2}\right)-\frac{2}{\Delta t}u^{n+1}$$
$$=-\frac{1}{h^2}\left(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}\left(\sin\phi\frac{\partial u^n}{\partial\phi}\right)+\left(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)}\right)\frac{\partial^2 u^n}{\partial\theta^2}\right)-\frac{2}{\Delta t}u^n\qquad(36)$$

Substitute (18) into (36), and from (34) we can derive

$$\frac{1}{h^2}\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}(\sin\phi\frac{\partial u_k^{n+1}(\phi)}{\partial\phi})-k^2(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)})u_k^{n+1}(\phi))e^{ik\theta}-\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\frac{2}{\Delta t}u_k^{n+1}(\phi)\,e^{ik\theta}$$

$$=-\frac{1}{h^2}\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}(\sin\phi\frac{\partial u_k^n(\phi)}{\partial\phi})-k^2(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)})u_k^n(\phi))e^{ik\theta}-\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}\frac{2}{\Delta t}u_k^n(\phi)\,e^{ik\theta}$$

$$=>\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}(\frac{1}{h^2}(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}(\sin\phi\frac{\partial u_k^{n+1}(\phi)}{\partial\phi})-k^2(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)})u_k^{n+1}(\phi))-\frac{2}{\Delta t}u_k^{n+1}(\phi))e^{ik\theta}$$

$$=\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}(-\frac{1}{h^2}(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}(\sin\phi\frac{\partial u_k^n(\phi)}{\partial\phi})-k^2(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)})u_k^n(\phi))-\frac{2}{\Delta t}u_k^n(\phi))e^{ik\theta}$$

Eliminate $\sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1}$ and $e^{ik\theta}$ from both sides of the equation,

$$\frac{1}{h^2}\left(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}\left(\sin\phi\frac{\partial u_k^{n+1}(\phi)}{\partial\phi}\right)-k^2\left(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)}\right)u_k^{n+1}(\phi)\right)-\frac{2}{\Delta t}u_k^{n+1}(\phi)$$
$$=-\frac{1}{h^2}\left(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}\left(\sin\phi\frac{\partial u_k^{n}(\phi)}{\partial\phi}\right)-k^2\left(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)}\right)u_k^{n}(\phi)\right)-\frac{2}{\Delta t}u_k^{n}(\phi)$$

After multiplying by h^2 to both sides of the equation and equaling the Fourier coefficients, $u_k(\phi)$ satisfies

$$\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}\left(\sin\phi\frac{\partial u_k^{n+1}(\phi)}{\partial\phi}\right)-k^2\left(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)}\right)u_k^{n+1}(\phi)-\frac{2h^2}{\Delta t}u_k^{n+1}(\phi)$$
$$=-\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}\left(\sin\phi\frac{\partial u_k^{n}(\phi)}{\partial\phi}\right)+k^2\left(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi)}\right)u_k^{n}(\phi)-\frac{2h^2}{\Delta t}u_k^{n}(\phi)$$

The Step 2 is equal to do FFT for θ as Figure 4.

Step 3: We use the symmetric discretization for ϕ, i.e.

$$\frac{\partial}{\partial\phi}\left(\sin\phi_i\frac{\partial U_i}{\partial\phi}\right)\approx\frac{\sin\phi_{i+\frac{1}{2}}(U_{i+1}-U_i)-\sin\phi_{i-\frac{1}{2}}(U_i-U_{i-1})}{\Delta\phi^2}$$

For each Fourier modes k ($k=-\frac{N}{2},-\frac{N}{2}+1,\ldots,0,1,\ldots,\frac{N}{2}-1.$), we set

$$u_k^{n+1}(\phi)=U^{n+1}\text{ at }T=(n+1)\Delta t$$
$$u_k^{n}(\phi)=U^{n}\text{ at }T=n\Delta t$$

and solve the following equations .

$$\frac{1}{\sin\phi_i}\frac{\sin\phi_{i+\frac{1}{2}}(U_{i+1}^{n+1}-U_i^{n+1})-\sin\phi_{i-\frac{1}{2}}(U_i^{n+1}-U_{i-1}^{n+1})}{\Delta\phi^2}-k^2\left(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi_i)}\right)U_i^{n+1}-\frac{2h^2}{\Delta t}U_i^{n+1}$$
$$=-\frac{1}{\sin\phi_i}\frac{\sin\phi_{i+\frac{1}{2}}(U_{i+1}^{n}-U_i^{n})-\sin\phi_{i-\frac{1}{2}}(U_i^{n}-U_{i-1}^{n})}{\Delta\phi^2}+k^2\left(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi_i)}\right)U_i^{n}-\frac{2h^2}{\Delta t}U_i^{n}$$

for $i=1,\ldots,M.$

Both sides of the equations were multiplied by $\Delta\phi^2\sin\phi_i$ and rearrange them,

$$\sin\phi_{i-\frac{1}{2}}U_{i-1}^{n+1}+\left(-\sin\phi_{i-\frac{1}{2}}-\sin\phi_{i+\frac{1}{2}}-k^2\Delta\phi^2\sin(\phi_i)\left(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi_i)}\right)-\frac{2h^2\Delta\phi^2\sin(\phi_i)}{\Delta t}\right)U_i^{n+1}$$
$$+\sin\phi_{i+\frac{1}{2}}U_{i+1}^{n+1}$$
$$=-\sin\phi_{i-\frac{1}{2}}U_{i-1}^{n}+\left(\sin\phi_{i-\frac{1}{2}}+\sin\phi_{i+\frac{1}{2}}+k^2\Delta\phi^2\sin(\phi_i)\left(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi_i)}\right)-\frac{2h^2\Delta\phi^2\sin(\phi_i)}{\Delta t}\right)U_i^{n}$$

$$-\sin\phi_{i+\frac{1}{2}}U_{i+1}^{n} \tag{37}$$

Therefore, we could set that

$$A_i = \sin\phi_{i-\frac{1}{2}}$$

$$B_i = -\sin\phi_{i-\frac{1}{2}} - \sin\phi_{i+\frac{1}{2}} - k^2\Delta\phi^2\sin(\phi_i)\left(\frac{1}{\sinh^2(\beta)} + \frac{1}{\sin^2(\phi_i)}\right) - \frac{2h^2\Delta\phi^2\sin(\phi_i)}{\Delta t}$$

$$C_i = \sin\phi_{i+\frac{1}{2}}$$

$$D_i = -B_i - \frac{4h^2\Delta\phi^2\sin(\phi_i)}{\Delta t}$$

for $i = 1, \dots, M$.

Because the coefficient of U_0 is $\sin\phi_{\frac{1}{2}} = 0$ and the coefficient of U_{M+1} is $\sin\phi_{M+\frac{1}{2}} = 0$, we have $A_1 = 0$ and $C_M = 0$. Therefore, we do not need symmetric constrain which is the great news for us to produce the following matrices,

$$\begin{bmatrix} B_1 & C_1 & & & \\ A_2 & B_2 & C_2 & & \\ & . & . & . & \\ & & A_{M-1} & B_{M-1} & C_{M-1} \\ & & & A_M & B_M \end{bmatrix} \begin{bmatrix} U_1^{n+1} \\ U_2^{n+1} \\ . \\ U_{M-1}^{n+1} \\ U_M^{n+1} \end{bmatrix}$$

$$= \begin{bmatrix} D_1 & -C_1 & & & \\ -A_2 & D_2 & -C_2 & & \\ & . & . & . & \\ & & -A_{M-1} & D_{M-1} & -C_{M-1} \\ & & & -A_M & D_M \end{bmatrix} \begin{bmatrix} U_1^{n} \\ U_2^{n} \\ . \\ U_{M-1}^{n} \\ U_M^{n} \end{bmatrix}$$

Because $A_{i+1} = \sin\phi_{i+\frac{1}{2}} = C_i$, for $i = 1, \dots, M-1$, the matrices will become symmetric matrices and lead to less computations. It reduces above matrices to be

$$\begin{bmatrix} B_1 & C_1 & & & \\ C_1 & B_2 & C_2 & & \\ & . & . & . & \\ & & C_{M-2} & B_{M-1} & C_{M-1} \\ & & & C_{M-1} & B_M \end{bmatrix} \begin{bmatrix} U_1^{n+1} \\ U_2^{n+1} \\ . \\ U_{M-1}^{n+1} \\ U_M^{n+1} \end{bmatrix}$$

$$= \begin{bmatrix} D_1 & -C_1 & & & \\ -C_1 & D_2 & -C_2 & & \\ & . & . & . & \\ & & -C_{M-2} & D_{M-1} & -C_{M-1} \\ & & & -C_{M-1} & D_M \end{bmatrix} \begin{bmatrix} U_1^{n} \\ U_2^{n} \\ . \\ U_{M-1}^{n} \\ U_M^{n} \end{bmatrix}$$

And these linear systems need to be solved N times by Thomas algorithm here.

The Step 3 is equal to solve the unknowns in the linear systems in each Fourier mode as Figure 5.

Step 4: Similar to Section 3.1.2 Step 4.

According to the procedure, i.e. Step 1=>Step 2=>Step 3=>Step 4, we could complete the numerical solution at one time step and we can derive the numerical solution at any time by keeping performing the procedure recursively.

4.2 Numerical Results

We could not find the exact solution of the equation (30), but yearn to make sure that our numerical solutions for the equation (30) are all right. It is a great challenge for us. There is one way to check if the solutions are right is comparing the solvers in Chapter 4 with Chapter 3 because we have the exact solution in Chapter 4.

4.2.1 Results of Solvers with the Central Difference Method

Compare Section 3.1.2 with Section 4.1.1, there are some changes in the components of their matrices that were made up by the Fourier modes of the linear systems in Section

3.1.2 as follows,

Mutating：

Spherical surface → Ellipsoid surface

$$R^2 \rightarrow h^2$$

$$\frac{k^2\Delta\phi^2}{\sin^2\phi_i} \rightarrow k^2\Delta\phi^2\left(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi_i)}\right)$$

$$B_i = -2-\frac{k^2\Delta\phi^2}{\sin^2(\phi_i)}-\frac{2R^2\Delta\phi^2}{\Delta t} \rightarrow B_i = -2-k^2\Delta\phi^2\left(\frac{1}{\sinh^2(\beta)}+\frac{1}{\sin^2(\phi_i)}\right)-\frac{2h^2\Delta\phi^2}{\Delta t}$$

$$D_i = -B_i-\frac{4R^2\Delta\phi^2}{\Delta t} \rightarrow D_i = -B_i-\frac{4h^2\Delta\phi^2}{\Delta t}$$

A_i and C_i are the same as before, for $i = 1, 2, \dots, M$. Modify our programming by above terms and then get the intuitional numerical solvers.

4.2.2 Results of Solvers with the Symmetric Discretization

Compare Section 3.1.3 with Section 4.1.2, there are some changes in the components of their matrices that were made up by the Fourier modes of the linear systems in Section 3.1.3 as follows,

Mutating：

Spherical surface → Ellipsoid surface

$$R^2 \rightarrow h^2$$

$$\frac{k^2\Delta\phi^2}{\sin(\phi_i)} \rightarrow \left(\frac{1}{\sinh^2\beta}+\frac{1}{\sin^2\phi_i}\right)\Delta\phi^2k^2\sin\phi_i$$

$$B_i = -A_i-C_i-\frac{1}{\sin\phi_i}\Delta\phi^2k^2-\frac{2R^2\Delta\phi^2\sin\phi_i}{\Delta t} \rightarrow B_i = -A_i-C_i-\left(\frac{1}{\sinh^2\beta}+\frac{1}{\sin^2\phi_i}\right)\Delta\phi^2k^2\sin\phi_i-\frac{2h^2\Delta\phi^2\sin\phi_i}{\Delta t}$$

$$D_i = -B_i-\frac{4R^2\Delta\phi^2\sin\phi_i}{\Delta t} \rightarrow D_i = -B_i-\frac{4h^2\Delta\phi^2\sin\phi_i}{\Delta t}$$

A_i and C_i are the same as before, for $i = 1, 2, \dots, M$. Modify our programming a little bit as above terms and then get the intuitional numerical solvers.

4.3 Mass Conservation

We developed an interest in keeping a lookout for the variation of the total mass in our numerical solvers due to (29),

here $\partial\Omega$: ellipsoid surface

$$x = \alpha\sinh\beta\sin\phi\cos\theta\,, y = \alpha\sinh\beta\sin\phi\sin\theta\,, z = \alpha\cosh\beta\cos\phi$$

$$\mathbf{X}(\alpha,\beta,\phi,\theta) = (x,y,z) = (\alpha\sinh\beta\sin\phi\cos\theta\,, \alpha\sinh\beta\sin\phi\sin\theta, \alpha\cosh\beta\cos\phi)$$

$$\mathbf{X}_{i,j} \approx \mathbf{X}\left(\phi_i,\theta_j\right)$$

$$\frac{\partial\mathbf{X}}{\partial\phi}\times\frac{\partial\mathbf{X}}{\partial\theta} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ \frac{\partial x}{\partial\phi} & \frac{\partial y}{\partial\phi} & \frac{\partial z}{\partial\phi} \\ \frac{\partial x}{\partial\theta} & \frac{\partial y}{\partial\theta} & \frac{\partial z}{\partial\theta} \end{vmatrix}$$

$$= \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ \alpha\sinh\beta\cos\phi\cos\theta & \alpha\sinh\beta\cos\phi\sin\theta & -\alpha\cosh\beta\sin\phi \\ -\alpha\sinh\beta\sin\phi\sin\theta & \alpha\sinh\beta\sin\phi\cos\theta & 0 \end{vmatrix}$$

$$= \left(\alpha^2\sinh\beta\cosh\beta\sin^2\phi\cos\theta, \alpha^2\sinh\beta\cosh\beta\sin^2\phi\sin\theta, \alpha^2\sinh^2\beta\sin\phi\cos\phi(\cos^2\theta+\sin^2\theta)\right)$$

$$= (\alpha^2\sinh\beta\cosh\beta\sin^2\phi\cos\theta, \alpha^2\sinh\beta\cosh\beta\sin^2\phi\sin\theta, \alpha^2\sinh^2\beta\sin\phi\cos\phi)$$

$$\left|\frac{\partial\mathbf{X}}{\partial\phi}\times\frac{\partial\mathbf{X}}{\partial\theta}\right| = (\alpha^4\sinh^2\beta\cosh^2\beta\sin^4\phi(\cos^2\theta+\sin^2\theta) + \alpha^4\sinh^4\beta\sin^2\phi\cos^2\phi)^{\frac{1}{2}}$$

$$= (\alpha^4\sinh^2\beta\cosh^2\beta\sin^4\phi + \alpha^4\sinh^4\beta\sin^2\phi\cos^2\phi)^{\frac{1}{2}}$$

$$= (\alpha^4\sinh^2\beta(1+\sinh^2\beta)\sin^4\phi + \alpha^4\sinh^4\beta\sin^2\phi\cos^2\phi)^{\frac{1}{2}}$$

$$= \left(\alpha^4\sinh^2\beta\sin^4\phi + \alpha^4\sinh^4\beta\sin^2\phi(\sin^2\phi+\cos^2\phi)\right)^{\frac{1}{2}}$$

$$= (\alpha^4\sinh^2\beta\sin^4\phi + \alpha^4\sinh^4\beta\sin^2\phi)^{\frac{1}{2}}$$

$$= \left(\alpha^4\sinh^2\beta\sin^2\phi(\sin^2\phi+\sinh^2\beta)\right)^{\frac{1}{2}}$$

$$= (\alpha^2\sinh^2\beta\sin^2\phi\, h^2)^{\frac{1}{2}}$$

$$= h\,\alpha\sinh\beta|\sin\phi|$$

Choose specific four cases in our paper to observe the total mass of them which changes with time. Because of immovability of our ellipsoid surface domain, the total mass is equal to

$$\int_0^{2\pi}\int_0^{\pi} u\left|\frac{\partial \mathbf{X}}{\partial \phi}\times\frac{\partial \mathbf{X}}{\partial \theta}\right| d\phi d\theta \approx \sum_{\substack{i=1,\ldots,M\\ j=1,\ldots,N}} u_{i,j}\left|\frac{\partial \mathbf{X}_{i,j}}{\partial \phi}\times\frac{\partial \mathbf{X}_{i,j}}{\partial \theta}\right| \Delta\phi\Delta\theta$$

$$= \sum_{\substack{i=1,\ldots,M\\ j=1,\ldots,N}} u_{i,j}\, h\, \alpha \sinh\beta |\sin\phi_i|\, \Delta\phi\Delta\theta$$

The total mass at n-th time step is set by

$$S^n \equiv \sum_{\substack{i=1,\ldots,M\\ j=1,\ldots,N}} u_{i,j}^n\, h\, \alpha \sinh\beta |\sin\phi_i|\, \Delta\phi\Delta\theta$$

The relative error at n-th time step in our numerical solver means

$$\frac{|S^n - S^0|}{S^0}$$

4.3.1 The Mass of Solvers with the Central Difference Method

We solved the heat equation (30) $u_t = \Delta_s u$ on an ellipsoid surface domain by numerical methods that are FFT, second-order central difference method 1, Crank-Nicolson method and Thomas algorithm.

Initial settings as Figure 36: $M = 32, N = 64,\ \alpha = 5,\ \beta = 0.5, \Delta t = \frac{1}{2\pi}\Delta x = \frac{1}{N}$.

(The ellipse rotating about z-axis has length $2\alpha\cosh\beta = 11.2762$ and the minor axis $2\alpha\sinh\beta = 5.221$.)

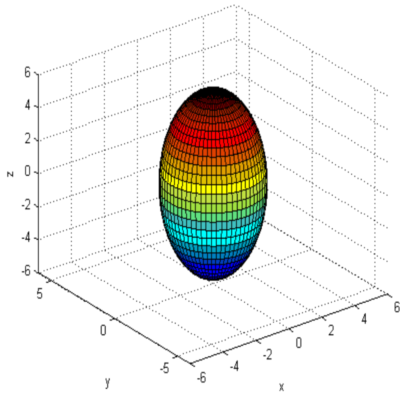


Figure 36: The domain is a ellipsoid surface with $\alpha = 5,\ \beta = 0.5$.

Case 1: Pizza-like initial condition as follows,

```
u(1:0.5*M,0.5*N:0.5*N+5)=1; others are set to be zero.
```

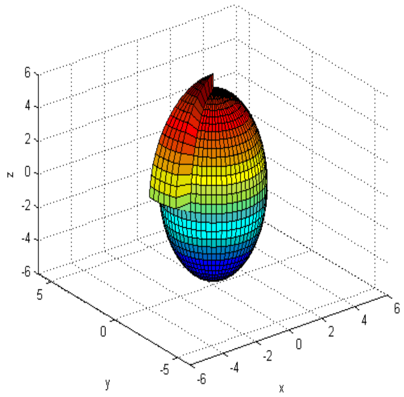


Figure 37: Initial value u; $T = 0$; Total mass $= 7.3208$.

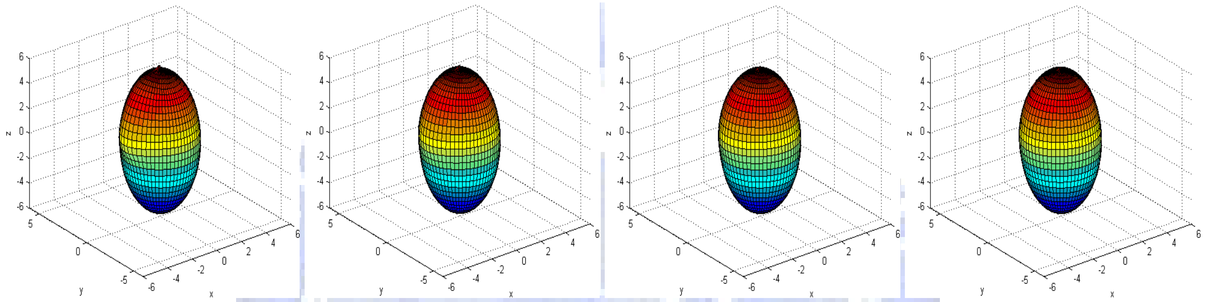


Figure 38: Left, $T = 0.25$, mass $= 7.3208$, relative error $= 7.2793\mathrm{e}-016$.
Mid-left, $T = 0.5$, mass $= 7.3208$, relative error $= 2.0625\mathrm{e}-015$.
Mid-right, $T = 0.75$, mass $= 7.3208$, relative error $= 9.7057\mathrm{e}-016$.
Right, $T = 0.1$, mass $= 7.3208$, relative error $= 2.4264\mathrm{e}-016$.

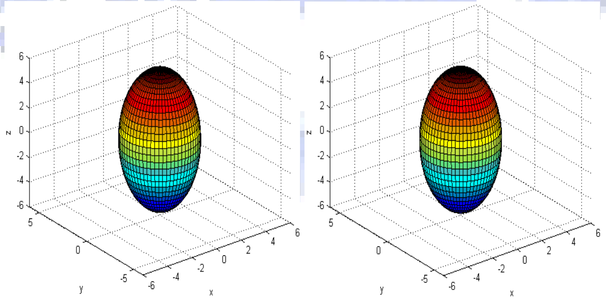


Figure 39: Left, $T = 5$, mass $= 7.3208$, relative error $= 8.3712\mathrm{e}-015$.
Right, $T = 20$, mass $= 7.3208$, relative error $= 2.4022\mathrm{e}-014$.

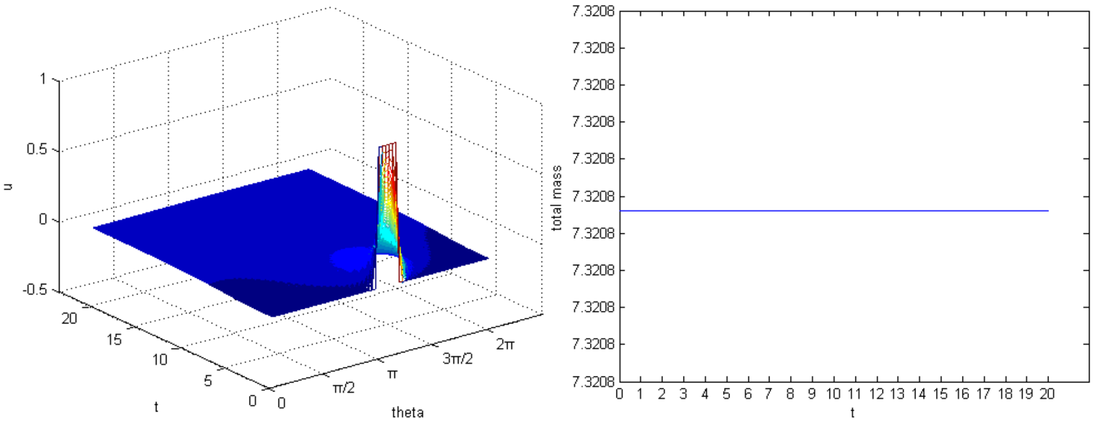


Figure 40: Left, the values of u on the equator change with time.
Right, total mass of u changes with time in $0\sim20$ seconds.

According to Figure 38, Figure 39, Figure 40, u was diffused as time goes by to be like the look of the original domain. Consider that total mass has little machine error. Therefore, Case 1 complies with the mass conservation law.

Case 2: Initial condition was set as one point as follows,

```
u(0.5*M,0.5*N+5)=1; others are set to be zero.
```

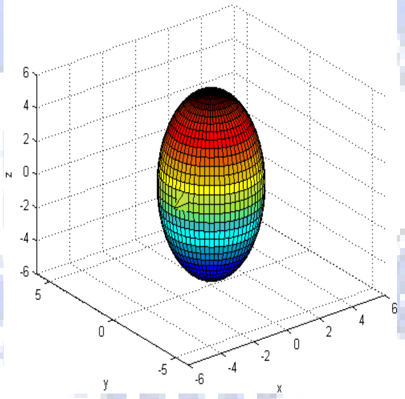


Figure 41: Initial value u; $T = 0$; $\text{Total mass} = 0.14128.$

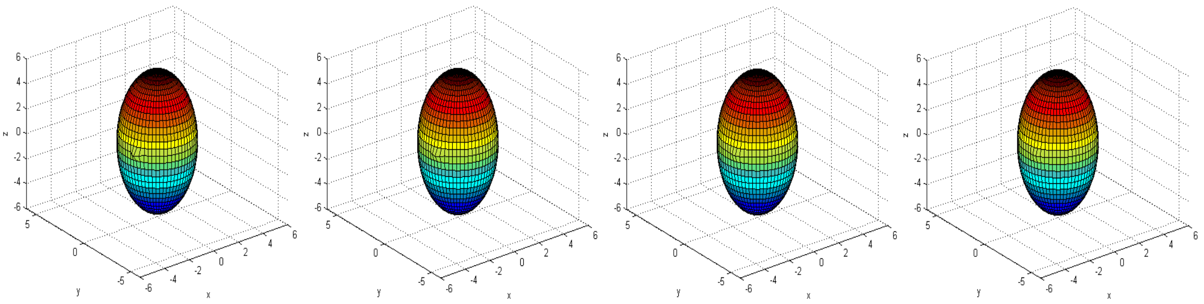


Figure 42: Left, $T = 1/64$, $\text{mass} = 0.14134$, $\text{relative error} = 0.00038824.$
Mid-left, $T = 1/32$, $\text{mass} = 0.14139$, $\text{relative error} = 0.000777.$
Mid-right, $T = 3/64$, $\text{mass} = 0.14145$, $\text{relative error} = 0.0011663.$

Right, $T = 1/16$, $\text{mass} = 0.1415$, $\text{relative error} = 0.001556$.

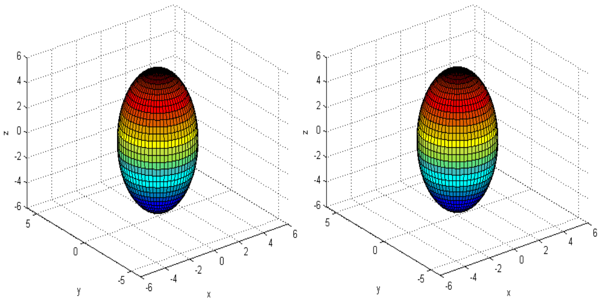


Figure 43： Left, $T = 5$, $\text{mass} = 0.15675$, $\text{relative error} = 0.10947$.

Right, $T = 20$, $\text{mass} = 0.16177$, $\text{relative error} = 0.14502$.

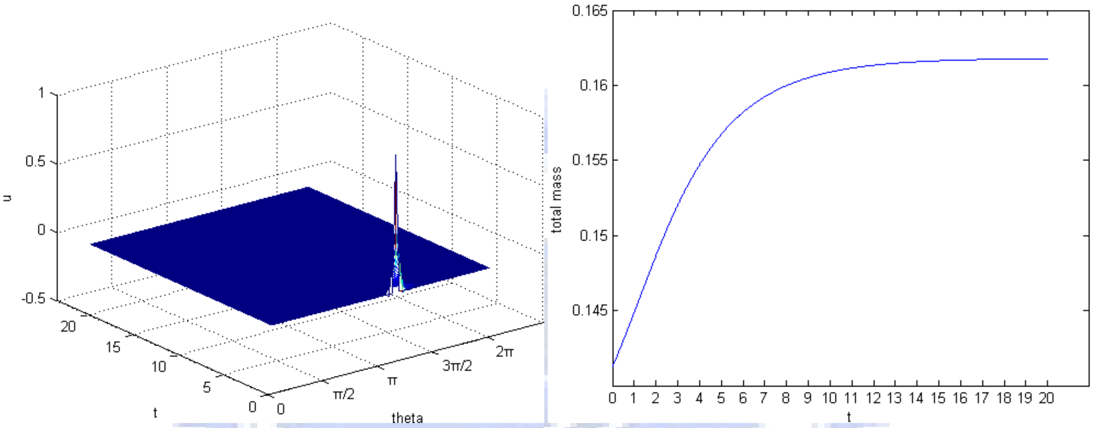


Figure 44： Left, the values of u on the equator change with time.

Right, total mass of u changes with time in $0\sim20$ seconds.

According to Figure 42, Figure 43, Figure 44, u was diffused as time goes by to be like the look of the original domain. Total mass is getting increasing, but it will turn to stable that means mass will not change any more in a long time. Case 2 does not comply with the mass conservation law.

Case 3: Chapeau-like initial condition as follows,

```
    u(1:3,1:N)=1; others are set to be zero.
```

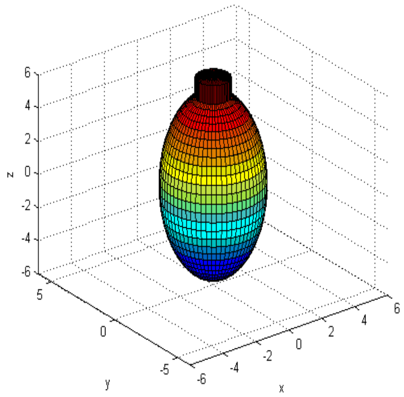


Figure 45: Initial value u; $T = 0$; Total mass $= 1.9674$.

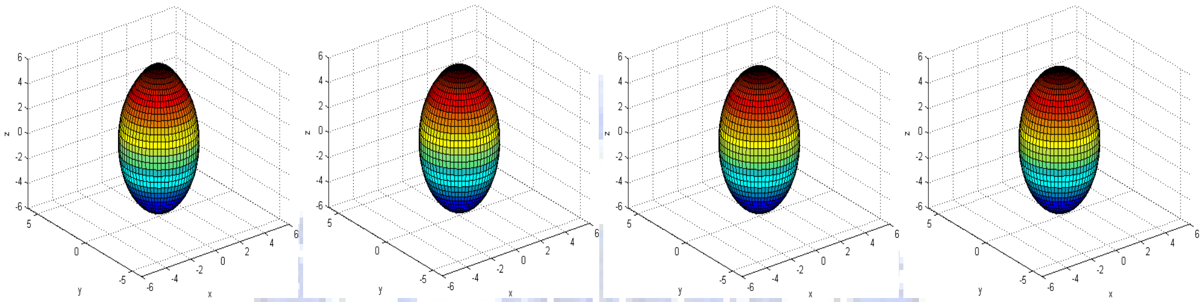


Figure 46: Left, $T = 0.25$, mass $= 1.7526$, relative error $= 0.10917$.
Mid-left, $T = 0.5$, mass $= 1.632$, relative error $= 0.17049$.
Mid-right, $T = 0.75$, mass $= 1.5505$, relative error $= 0.21191$.
Right, $T = 1$, mass $= 1.4902$, relative error $= 0.24256$.

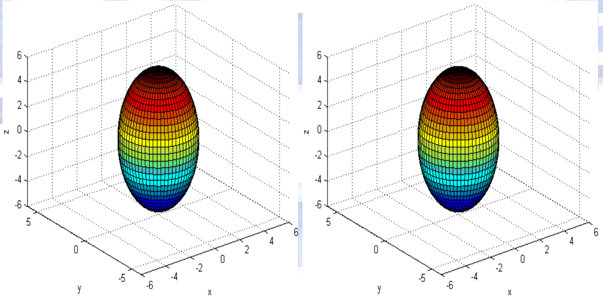


Figure 47: Left, $T = 5$, mass $= 1.192$, relative error $= 0.39413$.
Right, $T = 20$, mass $= 1.1192$, relative error $= 0.4311$.

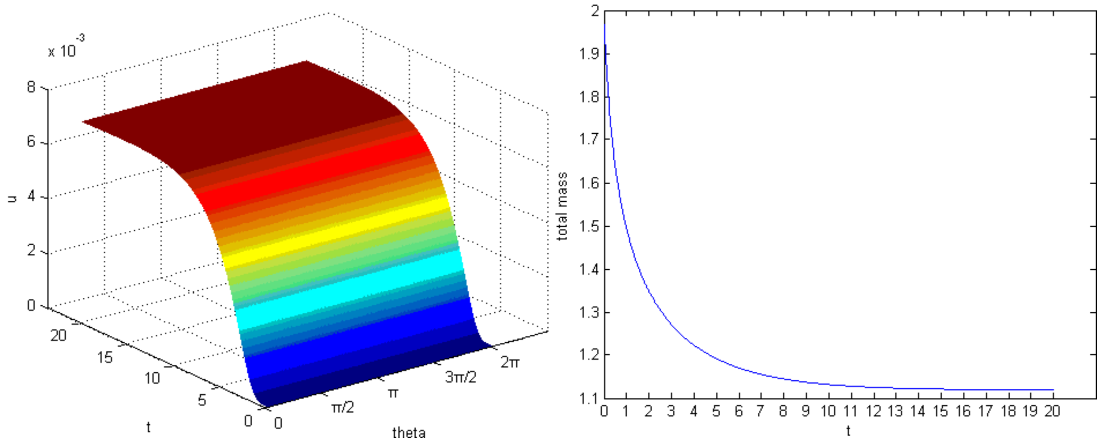


Figure 48: Left, the values of u on the equator change with time.
Right, total mass of u changes with time in $0\sim20$ seconds.

According to Figure 46, Figure 47, Figure 48, u was diffused as time goes by to be like the look of the original domain. Total mass is getting decreasing, but it will turn to stable that means mass will not change any more in a long time. Case 3 does not comply with the mass conservation law.

Case 4: Smooth initial condition as follows,

$$\mathtt{u(m,n)=abs(sin((m-0.5)*}\frac{\pi}{M}\mathtt{))+abs(cos((n-1)*}\frac{2\pi}{N}\mathtt{))},\quad \forall\, m,n.$$

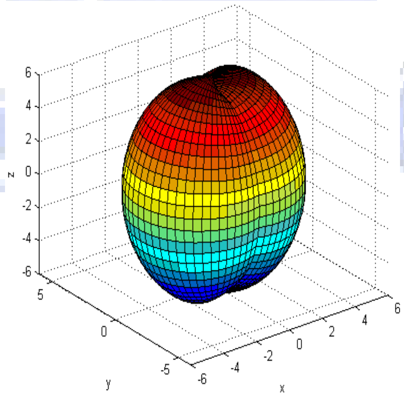


Figure 49: Initial value u; $T = 0$; $\text{Total mass} = 228.1165$.

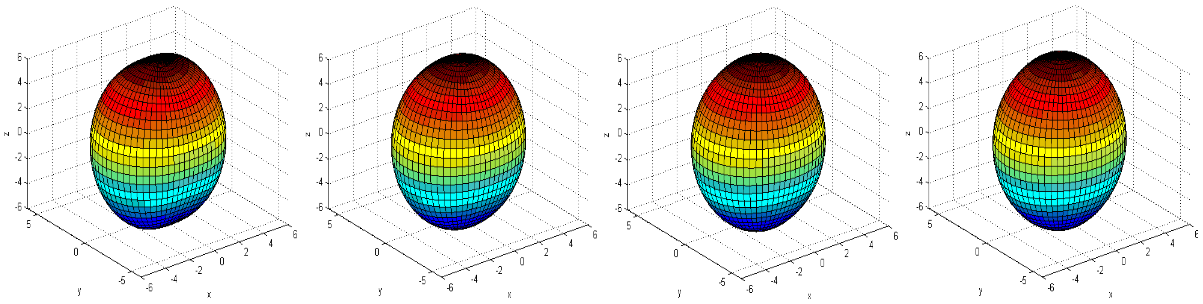


Figure 50： Left, $\mathrm{T} = 0.25$, mass $= 228.6505$, relative error $= 0.0023412$.
Mid-left, $\mathrm{T} = 0.5$, mass $= 229.088$, relative error $= 0.0042588$.
Mid-right, $\mathrm{T} = 0.75$, mass $= 229.4607$, relative error $= 0.0058928$.
Right, $\mathrm{T} = 1$, mass $= 229.7849$, relative error $= 0.0073137$.

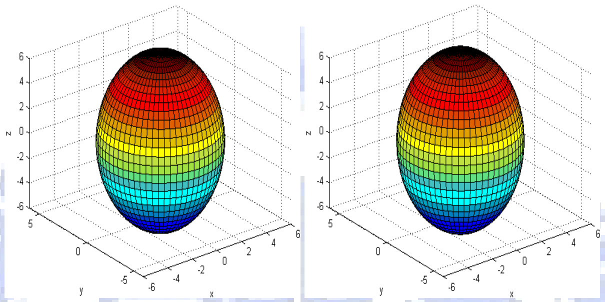


Figure 51： Left, $\mathrm{T} = 5$, mass $= 232.1409$, relative error $= 0.017642$.
Right, $\mathrm{T} = 20$, mass $= 232.8774$, relative error $= 0.02087$.

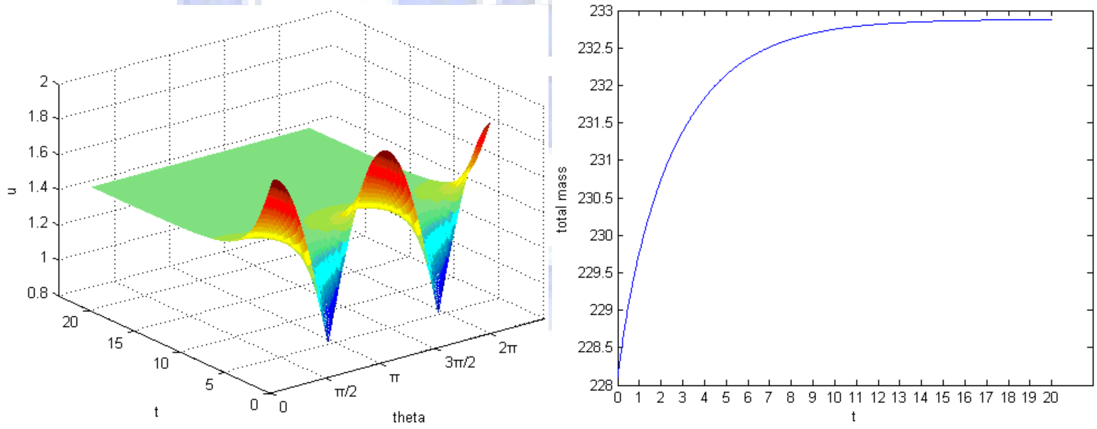


Figure 52： Left, the values of u on the equator change with time.
Right, total mass of u changes with time in $0 \sim 20$ seconds.

According to Figure 50, Figure 51, Figure 52, u was diffused as time goes by to be like the look of the original domain. Total mass is getting increasing, but it will turn to stable that means mass will not change any more in a long time. Case 4 does not comply with the mass conservation law.

In above Case 1~Case 4, we derive that Case 1 can get the perfect result by mass conservation with some machine error；the others show up big derivation in our relative error.

4.3.2 The Mass of solvers with the Symmetric Discretization

We solved the heat equation (34) $u_t = \Delta_s u$ on a ellipsoid surface domain by numerical methods that are FFT, symmetric discretization, Crank-Nicolson method and Thomas algorithm.

Initial settings as Figure 37: $M = 32, N = 64,\ \alpha = 5,\ \beta = 0.5, \Delta t = \frac{1}{2\pi}\Delta x = \frac{1}{N}$.

Case 1: Initial values are the same as Figure 36,

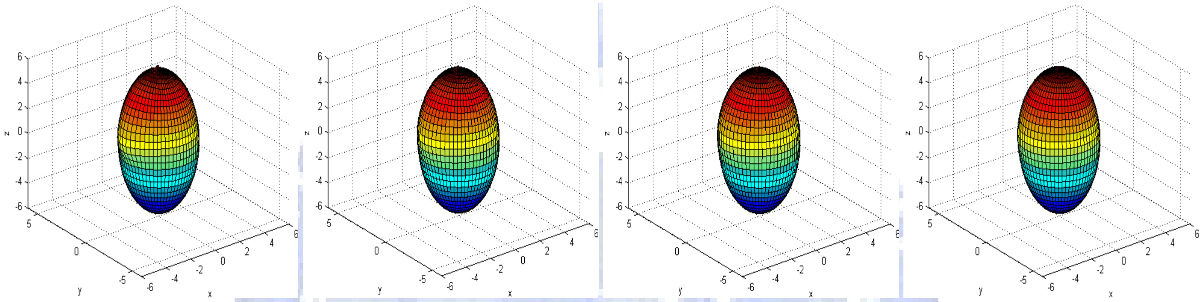


Figure 53: Left, $T = 0.25$, mass $= 7.3208$, relative error $= 2.4264e - 016$.
Mid-left, $T = 0.5$, mass $= 7.3208$, relative error $= 5.2168e - 015$.
Mid-right, $T = 0.75$, mass $= 7.3208$, relative error $= 2.0625e - 015$.
Right, $T = 0.1$, mass $= 7.3208$, relative error $= 1.9411e - 015$.

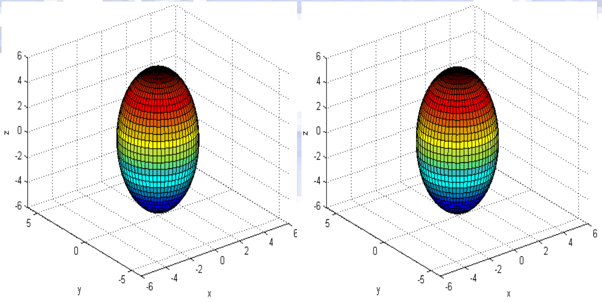


Figure 54: Left, $T = 5$, mass $= 7.3208$, relative error $= 1.189e - 014$.
Right, $T = 20$, mass $= 7.3208$, relative error $= 3.9187e - 014$.

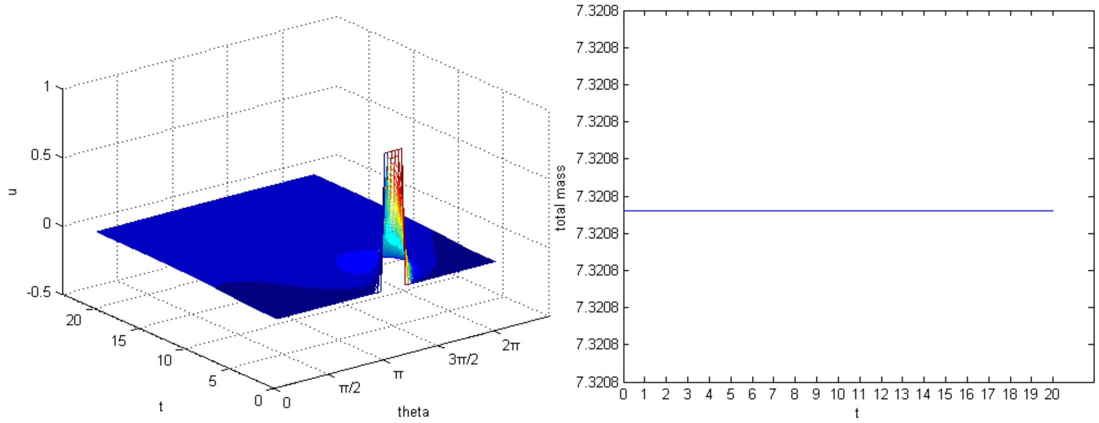


Figure 55: Left, the values of u on the equator change with time.
Right, total mass of u changes with time in $0\sim20$ seconds.

According to Figure 53, Figure 54, Figure 55, u was diffused as time goes by to be like the look of the original domain. Consider that total mass has little machine error. Therefore, Case 1 complies with the mass conservation law.

Case 2: Initial values are the same as Figure 41,

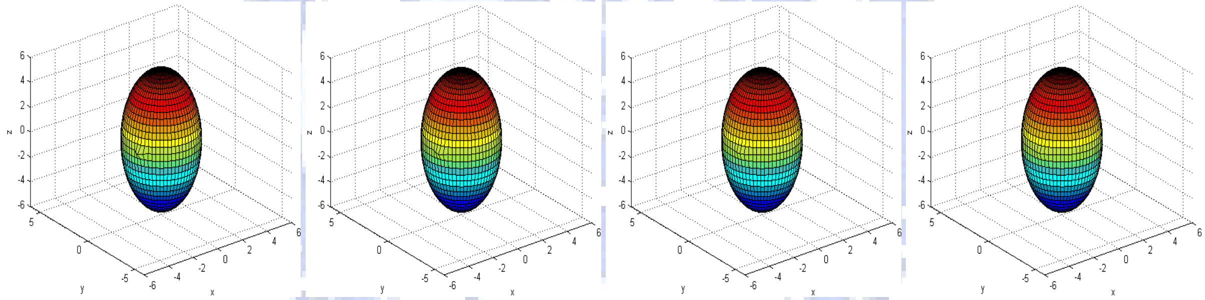


Figure 56: Left, $\mathrm{T} = 1/64$, $\mathrm{mass} = 0.14134$, $\mathrm{relative\ error} = 0.00038817$.
Mid-left, $\mathrm{T} = 1/32$, $\mathrm{mass} = 0.14139$, $\mathrm{relative\ error} = 0.00077686$.
Mid-right, $\mathrm{T} = 3/64$, $\mathrm{mass} = 0.14145$, $\mathrm{relative\ error} = 0.001166$.
Right, $\mathrm{T} = 1/16$, $\mathrm{mass} = 0.1415$, $\mathrm{relative\ error} = 0.0015557$.

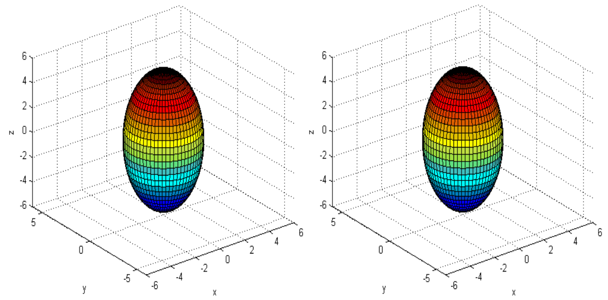


Figure 57: Left, $\mathrm{T} = 5$, $\mathrm{mass} = 0.15676$, $\mathrm{relative\ error} = 0.10953$.
Right, $\mathrm{T} = 20$, $\mathrm{mass} = 0.16179$, $\mathrm{relative\ error} = 0.14519$.

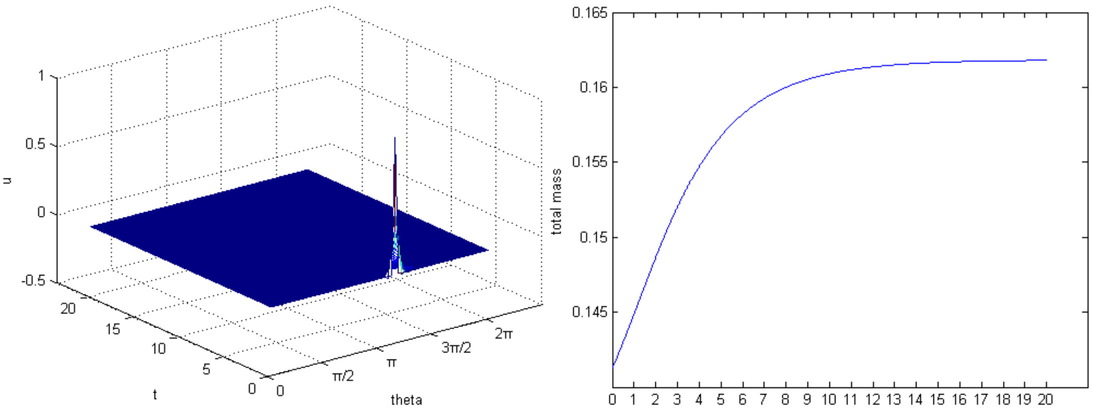


Figure 58: Left, the values of u on the equator change with time.
Right, total mass of u changes with time in 0~20 seconds.

According to Figure 56, Figure 57, Figure 58, u was diffused as time goes by to be like the look of the original domain. Total mass is getting increasing, but it will turn to stable that means mass will not change any more in a long time. Case 2 does not comply with the mass conservation law.

Case 3: Initial values are the same as Figure 45,

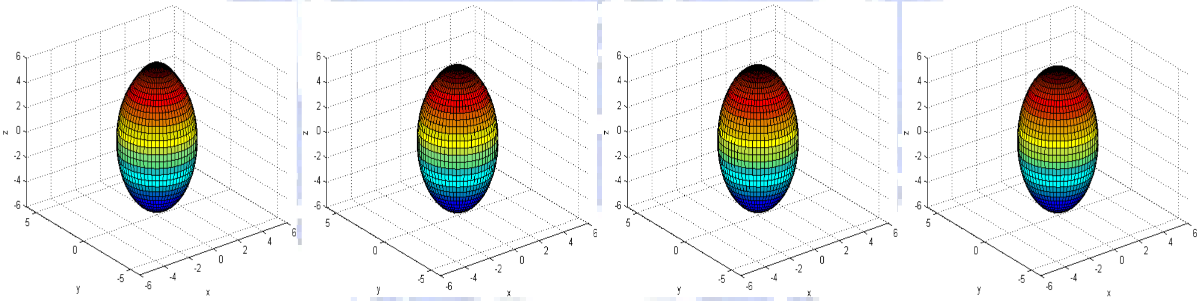


Figure 59: Left, $T = 0.25$, mass $= 1.752$, relative error $= 0.10944$.
Mid-left, $T = 0.5$, mass $= 1.6312$, relative error $= 0.17087$.
Mid-right, $T = 0.75$, mass $= 1.5496$, relative error $= 0.21235$.
Right, $T = 1$, mass $= 1.4892$, relative error $= 0.24304$.

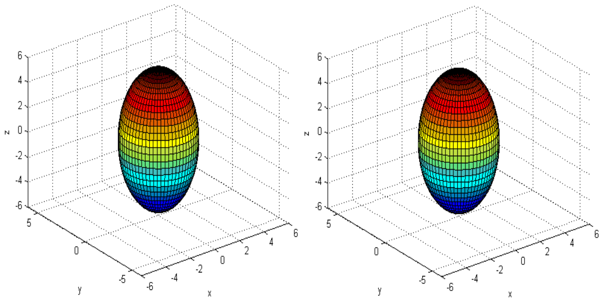


Figure 60: Left, $T = 5$, mass $= 1.1908$, relative error $= 0.39475$.
Right, $T = 20$, mass $= 1.1179$, relative error $= 0.43179$.

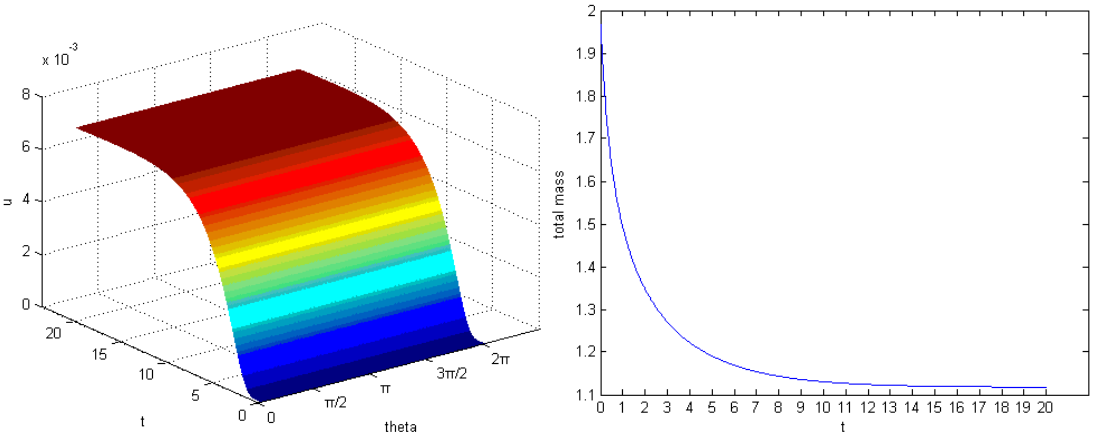


Figure 61： Left, the values of u on the equator change with time.
Right, total mass of u changes with time in 0~20 seconds.

According to Figure 59, Figure 60, Figure 61, u was diffused as time goes by to be like the look of the original domain. Total mass is getting decreasing, but it will turn to stable that means mass will not change any more in a long time. Case 3 does not comply with the mass conservation law.

Case 4: Initial values are the same as Figure 49,

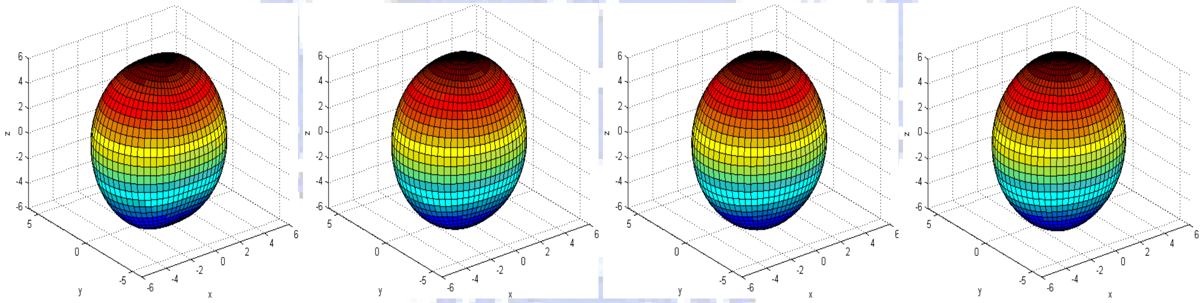


Figure 62： Left, $T = 0.25$, mass $= 228.651$, relative error $= 0.0023432$.
Mid-left, $T = 0.5$, mass $= 229.0887$, relative error $= 0.0042622$.
Mid-right, $T = 0.75$, mass $= 229.4618$, relative error $= 0.0058973$.
Right, $T = 1$, mass $= 229.7861$, relative error $= 0.0073193$.

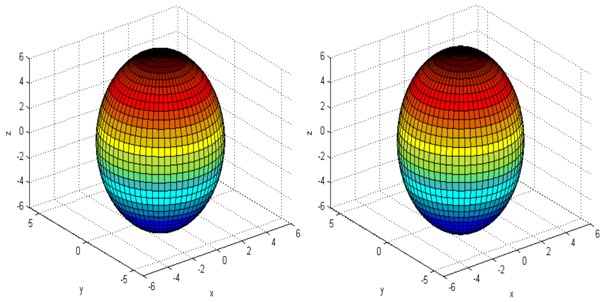


Figure 63： Left, $T = 5$, mass $= 232.1449$, relative error $= 0.017659$.

Right, $T = 20$, mass $= 232.8834$, relative error $= 0.020897$.

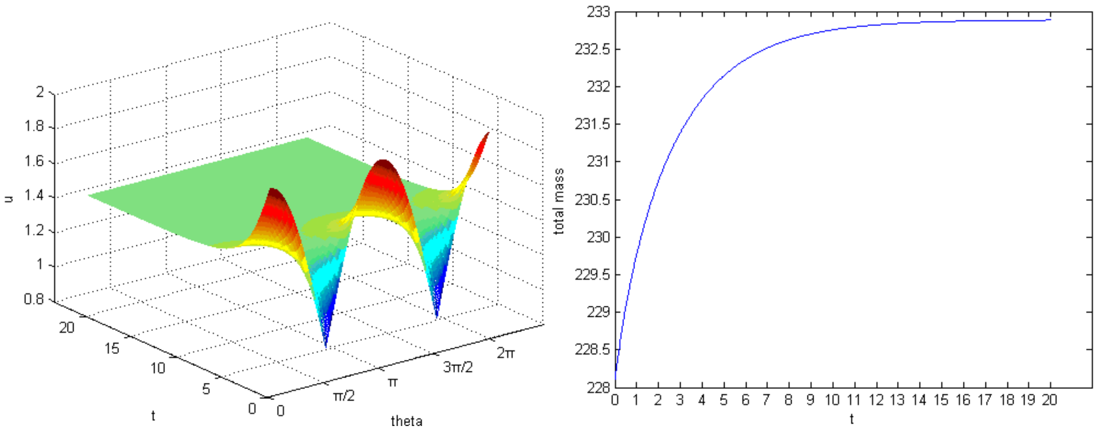


Figure 64: Left, the values of u on the equator change with time.
Right, total mass of u changes with time in $0\sim20$ seconds.

According to Figure 62, Figure 63, Figure 64, u was diffused as time goes by to be like the look of the original domain. Total mass is getting increasing a little bit, but it will turn to stable that means mass will not change any more in a long time. Case 4 does not comply with the mass conservation law.

In above Case 1~Case 4, we derive that Case 1 can get the perfect result by mass conservation with some machine error；the others show up big derivation in our relative error.

4.3.3 Comparison between the Central Difference Method and the Symmetric Discretization in Ellipsoid Coordinates

According to the results in Section 4.3.1 and Section 4.3.2, we could summarize the following table.

	4.3.1	**4.3.2**
Case1	2.4022e-014	3.9187e-014
Case2	0.14528	0.14519
Case3	0.4311	0.43179
Case4	0.02087	0.020897

Table 6: The relative error in different cases with different methods at $T = 20$.

Table 6 brings us that the solvers on the ellipsoid surface domain which use the second-order symmetric discretization or the solvers which use the second-order central difference method are coming to the same thing here.

4.3.4 Comparison between Chapter 3 and Chapter 4

After combining the results of Chapter 3 and the results of Chapter 4, we derive that the symmetry discretization which is based on the spherical surface domain will get perfect outcomes. Another result which we are interested in is that Case 1 complies with the mass conservation law both in Chapter 3 and Chapter 4. Because Chapter 4 have no exact solution, we plan to show the results of the case which is that take the ellipsoid surface domain to approximate to the spherical surface domain with the radius equals 5 as follows, set $\alpha = \frac{1}{40}$, $\beta = 6$, i.e. $\alpha\cosh\beta = 5.0429$, $\alpha\sinh\beta = 5.0428$,

	Spherical solver	Ellipsoid solver
Case1	2.8661e-013	1.3395e-014
Case2	0.00024238	0.00023811
Case3	0.001145	0.001137
Case4	5.0161e-005	4.9565e-005

Table 7: The relative error in different cases with the central difference method at $T = 20$.

	Spherical solver	Ellipsoid solver
Case1	1.1455e-014	1.8136e-014
Case2	4.9591e-015	4.0326e-006
Case3	6.563e-016	7.6251e-006
Case4	3.3088e-015	5.6232e-007

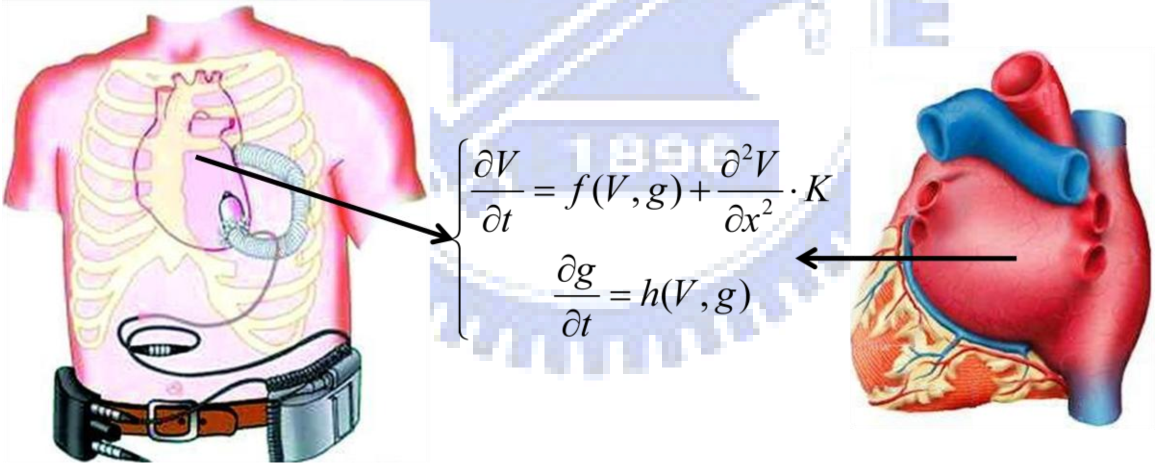
Table 8: The relative error in different cases with symmetric discretization at $T = 20$.

In Table 8, symmetric discretization makes perfect results in spherical coordinates, and central difference makes good results in ellipsoid coordinates. When we ignore the efficiency of the symmetric discretization and observe Table 7, they seem very close. According to above two tables, we declare that our solvers of Chapter 4 are correct, and the total mass is concerned with the geometry of the domain or the configuration of our grids because the sphere and ellipsoid have dense grids near the poles and dissipative grids near the equator.

5、The Primary Cardiac Simulation of Human Beings

5.1 The Reaction-Diffusion Equation

The equation that we want to solve comes from [15], and we will really solve the system which is in [6]. Because some action potentials of several kinds of cells in the healthy heart can now be calculated from the following equations of ion flow and voltage change as Figure 65 [15].



$$\begin{cases} \dfrac{\partial V}{\partial t} = f(V,g) + \dfrac{\partial^2 V}{\partial x^2} \cdot K \\ \dfrac{\partial g}{\partial t} = h(V,g) \end{cases}$$

Figure 65：V：membrane voltage; f：ionic current; K：physical factor; g：openness of ion channel or channel conductivities; h：a mathematical function of present openness and local potential[15].

V is the local electric potential which changes with time at a rate composed of two functions. One function have relations with the geometric arrangement of neighboring potentials; it equals the electronic currents that couple membranes across space. K is the same as a chemical diffusion coefficient. f is a local ionic current. g is the local openness of each ion channel. H： a mathematical function of present openness and local potential[15]. Through the strict interpretations of [15], we got the meanings of all variables. Here we get the meanings of the variables of the reaction-diffusion systems in [6]. The reaction-diffusion systems in [6] come from [2] as following discussion. We start with a two-variable system of reaction -diffusion equations modeling the dynamics of an excitable medium：

$$\begin{cases} \frac{\partial \mathrm{u}}{\partial \mathrm{t}} = \mathrm{f}(\mathrm{u},\mathrm{v}) + \Delta \mathrm{u} \\ \frac{\partial \mathrm{v}}{\partial \mathrm{t}} = \mathrm{g}(\mathrm{u},\mathrm{v}) \end{cases} \tag{38}$$

where the functions $\mathrm{f}(\mathrm{u},\mathrm{v})$ and $\mathrm{g}(\mathrm{u},\mathrm{v})$ represent the local kinetics of the two variables u and v. Choose the diffusion coefficient for $\Delta\mathrm{u}$ be unity and model the local kinetics with the equations

$$\mathrm{f}(\mathrm{u},\mathrm{v}) = \frac{1}{\varepsilon}\mathrm{u}(1-\mathrm{u})[\mathrm{u}-\mathrm{u}_{\mathrm{th}}(\mathrm{v})] \;\;,\;\; \mathrm{g}(\mathrm{u},\mathrm{v}) = \mathrm{u}-\mathrm{v}$$

$$\mathrm{u}_{\mathrm{th}}(\mathrm{v}) = \frac{(\mathrm{v}+\mathrm{b})}{\mathrm{a}}$$

and a, b, ε are parameters. Set ε to be small, so as to make u be very fast. The variables u and v are known as the excitation and recovery variables, respectively. After combining the equations in [2] with the equations in [15], we derive the meaning of each variable and function:

u: Local electric potential.

v: Local openness of each ion channel.

f: Local ionic current.

g: The rates of change of openness of each ion channel concern with present openness and local potential.

5.2 Numerical Methods and Techniques

5.2.1 Gilbert Strang Splitting Method

Solve directly the equation that has both linear terms and nonlinear terms is very difficult for us. It is a good news for us that the equation could be separated into two parts directly as follows:

$$u_t = f + g \tag{39}$$

here u, f, g are functions, and f is a nonlinear term and g is a linear term. The Gilbert Strang Splitting method is good for these kinds of problem.

Step 1: Solve $u_t = g$ at $\frac{1}{2}\Delta t$ time step.

Step 2: Solve $u_t = f$ at Δt time step.

Step 3: Solve $u_t = g$ at $\frac{1}{2}\Delta t$ time step.

The procedure of Step 1 => Step 2 => Step 3 is equal to solve equation (39) at Δt time step. If we could find the exact solution for $u_t = g$ (Because it is more possible to make it in linear term for us.), then the solution will be more accurate.

5.2.2 Numerical Techniques

We introduce the numerical techniques in [6] to solve the governing equation (38). Beacause the system spends most of the time within the small boundary layer δ, the way to solve is that let $u \neq 0$ where outside the boundary layer part of the wave; otherwise $u = 0$. We add the process by means of the following algorithm [2]:

If $u^n < \delta$, then

$$u^{n+1} = 0,\ v^{n+1} = (1 - \Delta t)v^n$$

Otherwise,

$$u_{th} = \frac{v^n + b}{a}$$

$$v^{n+1} = v^n + \Delta t(u^n - v^n)$$

$$u^{n+1} = u^n + \left({}^{\Delta t}/_{\varepsilon}\right)u^n(1 - u^n)(u^n - u_{th})$$

where u^n and v^n are the values of the u and v at the n-th time step. By above idea, the reaction term can be more efficient with this little effort in time steps, because most of the spatial points are within a small boundary layer δ at any instant of the time.

5.3 Numerical Solvers

We want to solve the reaction-diffusion equation as follows,

$$\begin{cases} \begin{cases} \dfrac{\partial u}{\partial t} = f(u,v) + \Delta u \\ \dfrac{\partial v}{\partial t} = g(u,v) \end{cases} \\ f(u,v) = \dfrac{1}{\varepsilon} u(1-u)[u - u_{th}(v)]\, , g(u,v) = u - v \\ u_{th}(v) = \dfrac{(v+b)}{a} \end{cases} \tag{40}$$

where $u = u(\phi, \theta, t)$ and $v = v(\phi, \theta, t)$, where $0 \le \phi \le \pi$, $0 \le \theta \le 2\pi$.

The periodic conditions

$$u(\phi, 0, t) = u(\phi, 2\pi, t), \ \forall\, t.$$

that we will use are inherent in the problem.

5.3.1 Numerical Solver of the Reaction–Diffusion Equation on a Spherical Surface Domain

The surface Laplacian on a spherical surface domain can describe as (22)

$$\Delta_s u = \nabla_s^2 u = \frac{1}{r^2}\left(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}\left(\sin\phi\frac{\partial u}{\partial\phi}\right) + \frac{1}{\sin^2(\phi)}\frac{\partial^2 u}{\partial\theta^2}\right) \tag{41}$$

Perform a numerical solver of the systems with (41) here. We use the Section 5.2.1 Gilbert Strang Splitting method to deal with the systems and combine time splitting and spectrum method (FFT) with finite difference methods as Section 3.1.3.

Step 1: Solve $\frac{\Delta t}{2}$ time step

$$\begin{cases} \dfrac{\partial u}{\partial t} = \dfrac{1}{\varepsilon} u(1-u)\left[u - \dfrac{(v+b)}{a}\right] \\ \dfrac{\partial v}{\partial t} = u - v \end{cases} \tag{42}$$

with explicit Forward-Euler method and impose Section 5.2.2 on it.

After (42) was discretized,

$$\begin{cases} \dfrac{u^{n+1}-u^{n}}{\Delta t/2} = \dfrac{1}{\varepsilon}u^{n}(1-u^{n})\left[u^{n} - \dfrac{(v^{n}+b)}{a}\right] \\ \dfrac{v^{n+1}-v^{n}}{\Delta t/2} = u^{n} - v^{n} \end{cases} \tag{43}$$

Impose Section 5.2.2 to (43),

If $u^{n} < \delta$, then

$$u^{n+1} = 0 \ , v^{n+1} = \left(1 - \frac{\Delta t}{2}\right) v^{n}$$

Otherwise,

$$u^{n+1} = u^{n} + \frac{\Delta t}{2\varepsilon}u^{n}(1-u^{n})\left(u^{n} - \frac{v^{n}+b}{a}\right)$$

$$v^{n+1} = v^{n} + \frac{\Delta t}{2}(u^{n} - v^{n})$$

Step 2: Take u^{n+1} and v^{n+1} of Step 1 be u^{n} and v^{n} in this step and solve Δt time step

$$\begin{cases} \dfrac{\partial u}{\partial t} = \Delta_s u = \dfrac{1}{r^2}\left(\dfrac{1}{\sin\phi}\dfrac{\partial}{\partial\phi}\left(\sin\phi\dfrac{\partial u}{\partial\phi}\right) + \dfrac{1}{\sin^2(\phi)}\dfrac{\partial^2 u}{\partial\theta^2}\right) \\ \dfrac{\partial v}{\partial t} = 0 \end{cases}$$

to derive new u^{n+1} and v^{n+1}. It is easy to know that we only need to solve

$$\frac{\partial u}{\partial t} = \frac{1}{r^2}\left(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}\left(\sin\phi\frac{\partial u}{\partial\phi}\right) + \frac{1}{\sin^2(\phi)}\frac{\partial^2 u}{\partial\theta^2}\right)$$

Above equation had been solved by Section 3.1.3.

Step 3: Take u^{n+1} and v^{n+1} of Step 2 be u^{n} and v^{n} in this step and solve $\frac{\Delta t}{2}$ time step as Step 1.

According to the procedure, i.e. Step 1=>Step 2=>Step 3, we could complete the numerical solver of the governing equation (40) on a spherical surface at one time step and we can derive the numerical solution at any time by keeping performing the procedure recursively.

5.3.2 Numerical Solver of the Reaction–Diffusion Equation on an Ellipsoid Surface Domain

The surface Laplacian on an ellipsoid surface domain can describe as (34)

$$\Delta_s u = \nabla_s^2 u = \frac{1}{h^2}\left(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}\left(\sin\phi\frac{\partial u}{\partial\phi}\right) + \left(\frac{1}{\sinh^2(\rho)} + \frac{1}{\sin^2(\phi)}\right)\frac{\partial^2 u}{\partial\theta^2}\right) \tag{44}$$

Perform a numerical solver of the systems with (44) here. We use the Section 5.2.1 Gilbert Strang Splitting method to deal with this systems and combine time splitting and spectrum method (FFT) with finite difference methods as Section 4.1.3.

Step 1: Be the same as Section 5.3.1 Step 1.

Step 2: Take u^{n+1} and v^{n+1} of Step 1 be u^n and v^n in this step and solve Δt time step

$$\begin{cases} \dfrac{\partial u}{\partial t} = \Delta_s u = \dfrac{1}{h^2}\left(\dfrac{1}{\sin\phi}\dfrac{\partial}{\partial\phi}\left(\sin\phi\dfrac{\partial u}{\partial\phi}\right) + \left(\dfrac{1}{\sinh^2(\beta)} + \dfrac{1}{\sin^2(\phi)}\right)\dfrac{\partial^2 u}{\partial\theta^2}\right) \\ \qquad\qquad \dfrac{\partial v}{\partial t} = 0 \end{cases}$$

to derive new u^{n+1} and v^{n+1}. It is easy to know that we only need to solve

$$\frac{\partial u}{\partial t} = \frac{1}{h^2}\left(\frac{1}{\sin\phi}\frac{\partial}{\partial\phi}\left(\sin\phi\frac{\partial u}{\partial\phi}\right) + \left(\frac{1}{\sinh^2(\beta)} + \frac{1}{\sin^2(\phi)}\right)\frac{\partial^2 u}{\partial\theta^2}\right)$$

Above equation had been solved by Section 4.1.3.

Step 3: Be the same as Section 5.3.1 Step 3.

According to the procedure, i.e. Step 1=>Step 2=>Step 3, we could complete the numerical solver of the governing equation (40) on a ellipsoid surface at one time step and we can derive the numerical solution at any time by keeping performing the procedure recursively.

5.4 Results and Analysis

In this section, we simulate the reaction-diffusion equation (40) on a spherical surface domain by setting $R = 16$ in Section 5.4.1. By comparing our numerical solver with the results in [6] to make sure that our solver is far and away right. Furthermore, we simulate the reaction-diffusion equation (40) on a ellipsoid surface domain which approximate the spherical surface domain with $R = 16$ in Section 5.4.2. After comparing Section 5.4.1 and Section 5.4.2 with [6], we know that they all have the same spiral wave. We could make sure that our solvers are correct. Therefore, solve (40) on the ellipsoid surface domain with cardiac size which we approximate in Section 5.4.3.

5.4.1 Numerical Results of the Reaction–Diffusion Equation on a Spherical Surface Domain with R=16

From [6], set $M \times N = 60 \times 120$, $R = 16$, $\Delta t = \frac{1}{2\pi}\Delta x = \frac{1}{N}$, $a = 0.35$, $b = 0.0008$, $\varepsilon = 0.02$, $\delta = 0.00001$ [3]. We only show and care about the values of u but ignore v, because the motions of v are not important for us. In our MATLAB, given initial values

```
u(1:M/2,M:M+8)=0.9; others are set to be zero.
v(1:M/2,M+8:M+16)=0.6; others are set to be zero.
```

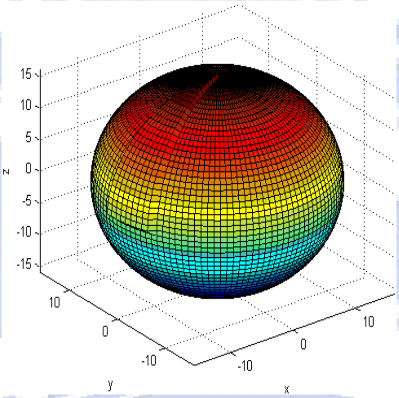


Figure 66: The initial values of u on a spherical surface with $R = 16$.

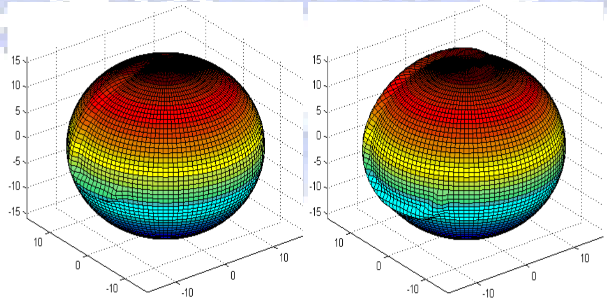


Figure 67: Left, $T = 1$. Right, $T = 2$.

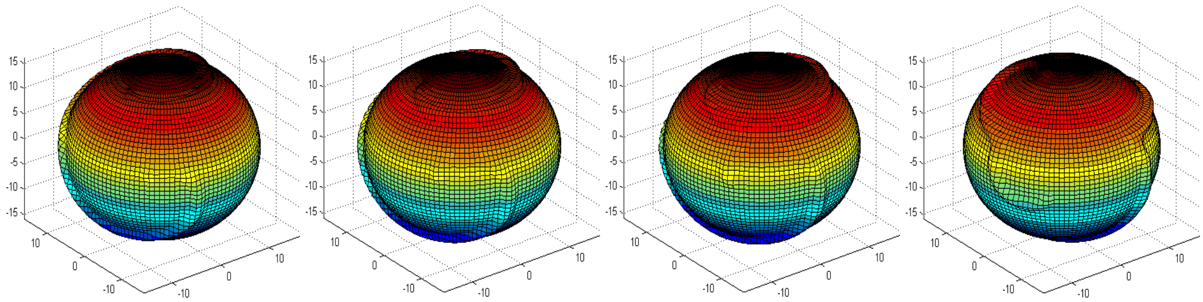


Figure 68: Left, $T = 9.5$. Mid-left, $T = 10$. Mid-right, $T = 10.5$. Right, $T = 12$.

Compare these results with [6], we could say that our numerical solution is correct.

5.4.2 Numerical Results of the Reaction–Diffusion Equation on an Ellipsoid Surface Domain which Approximates to R=16

Replace $\mathrm{R} = 16$ by $\alpha = 1.6$, $\beta = 3$ in Section 5.4.1, and it leads to $\alpha\cosh(\beta) = 16.1083$, $\alpha\sinh(\beta) = 16.0286$.

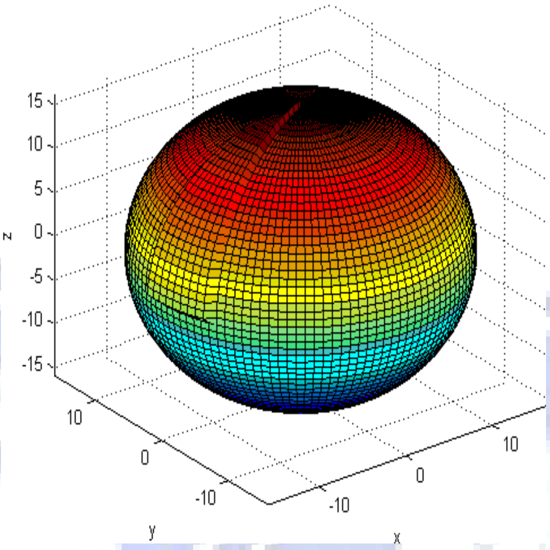


Figure 69: The initial values of u on a ellipsoid surface with $\alpha\cosh(\beta) = 16.1083$ and $\alpha\sinh(\beta) = 16.0286$.

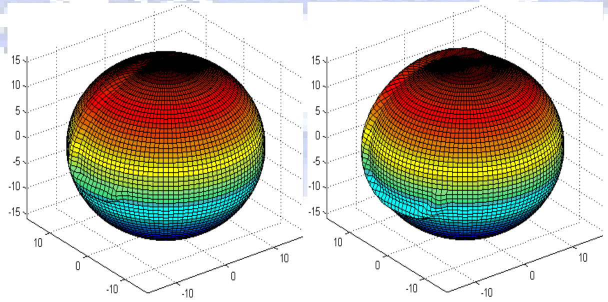


Figure 70: Left, $\mathrm{T} = 1$. Right, $\mathrm{T} = 2$.

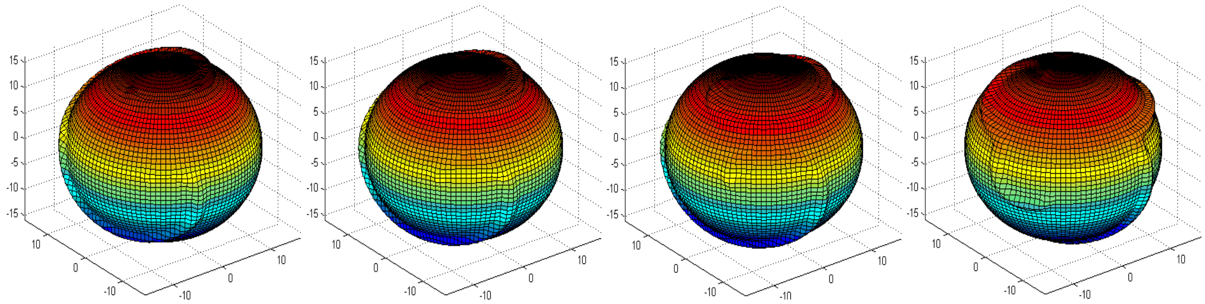


Figure 71: Left, $\mathrm{T} = 9.5$. Mid-left, $\mathrm{T} = 10$. Mid-right, $\mathrm{T} = 10.5$. Right, $\mathrm{T} = 12$.

Compare these results with Section 5.4.1, we could say that our numerical solution on an ellipsoid domain is correct and the solver in Chapter 4 is right.

5.4.3 Numerical results of the Reaction–Diffusion Equation on an Ellipsoid Surface Domain which Approximates to Cardiac Size

Replace $\alpha = 1.6$, $\beta = 3$, $\varepsilon = 0.02$ by $\alpha = 0.35$, $\beta = 0.0008$, $\varepsilon = 0.025$ in Section 5.4.2, and it leads to $\alpha\cosh(\beta) = 6.9935$, $\alpha\sinh(\beta) = 5.0094$. In our perennial tests, finally we take $\varepsilon = 0.025$ to let the results be periodic. Cardiac size is like a fist about $14\text{cm} \times 10\text{cm}$ in our settings.

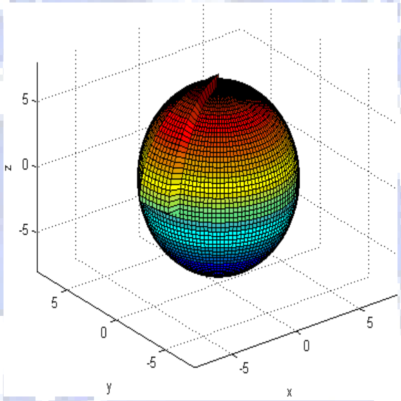


Figure 72: The initial values of u on a ellipsoid surface with $\alpha\cosh(\beta) = 6.9935$ and $\alpha\sinh(\beta) = 5.0094$.

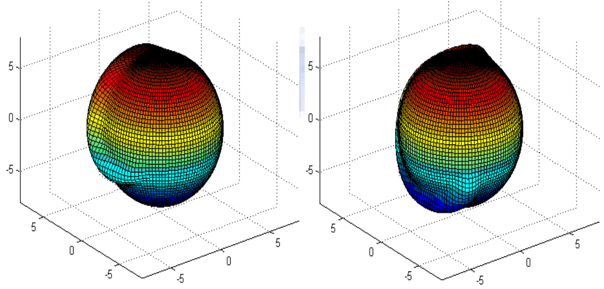


Figure 73: Left, $T = 1$. Right, $T = 2$.

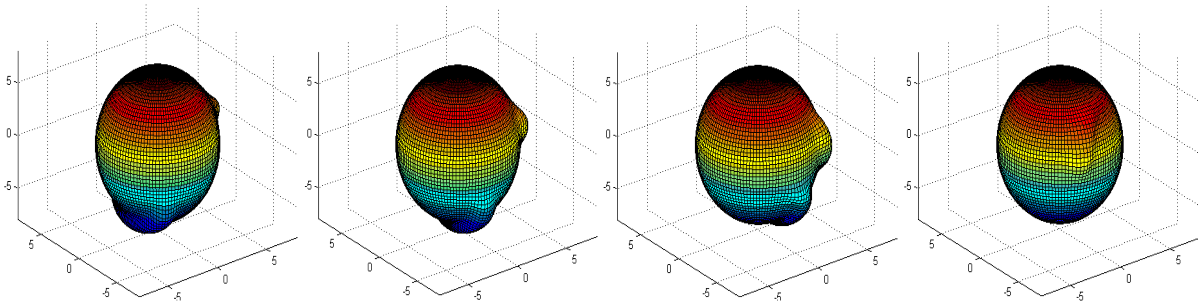


Figure 74: Left, T = 9.5. Mid-left, T = 10. Mid-right, T = 10.5. Right, T = 12.

Further, observe the values in the future and they are periodic by the frequency analysis.

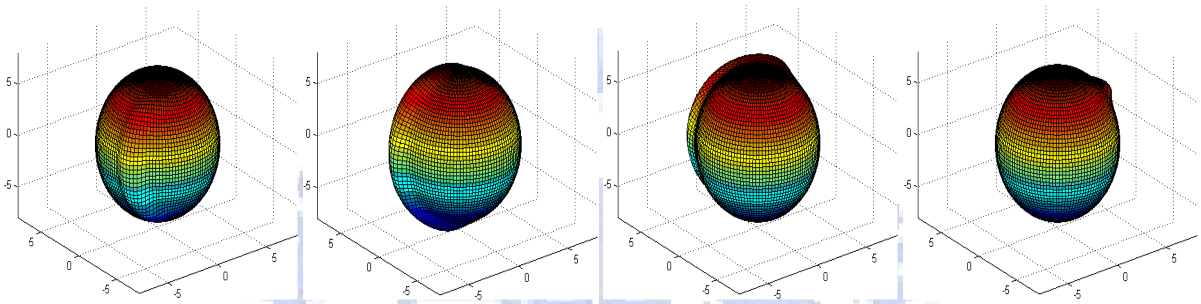


Figure 75: Left, T = 20. Mid-left, T = 21. Mid-right, T = 22. Right, T = 23.

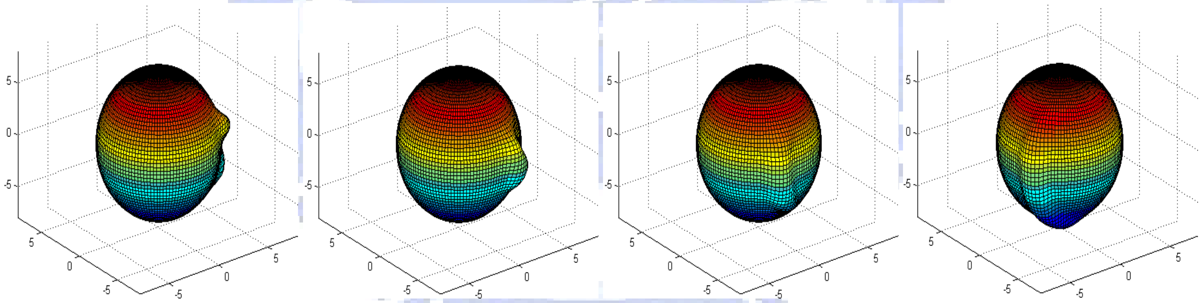


Figure 76: Left, T = 24. Mid-left, T = 25. Mid-right, T = 26. Right, T = 27.

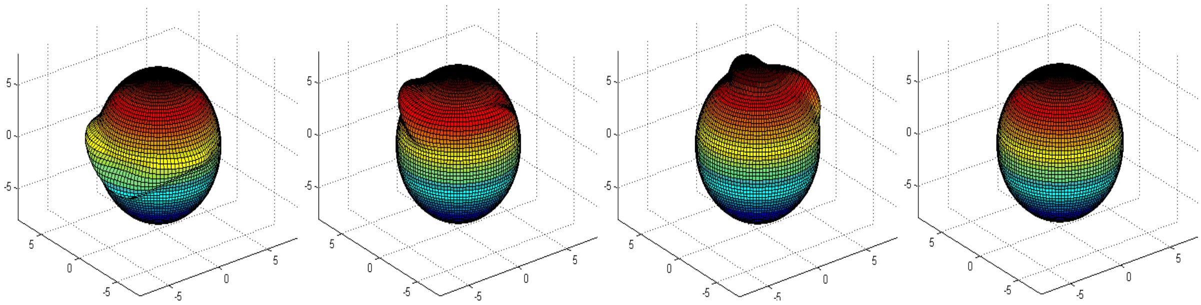


Figure 77: Left, T = 520. Mid-left, T = 521. Mid-right, T = 522. Right, T = 523.

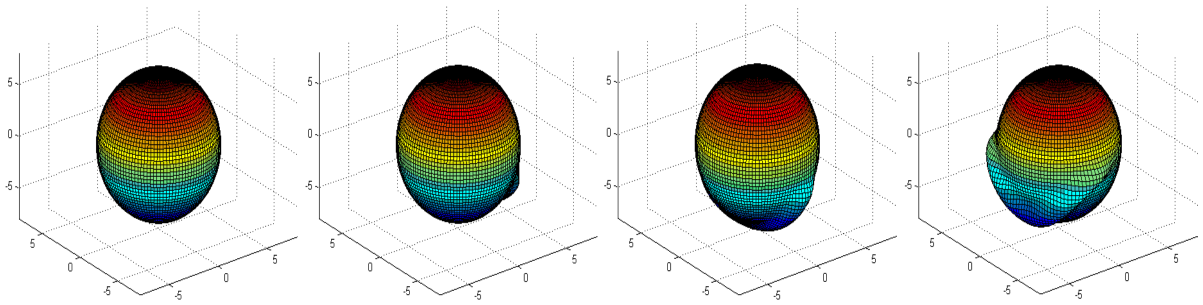


Figure 78: Left, T = 524. Mid-left, T = 525. Mid-right, T = 526. Right, T = 527.

From Figure 75 to Figure 76 and from Figure 77 to Figure 78, it is easy to see that the wave is periodic about 7~8 seconds. Pick representative fourteen points to observe the periodicity.

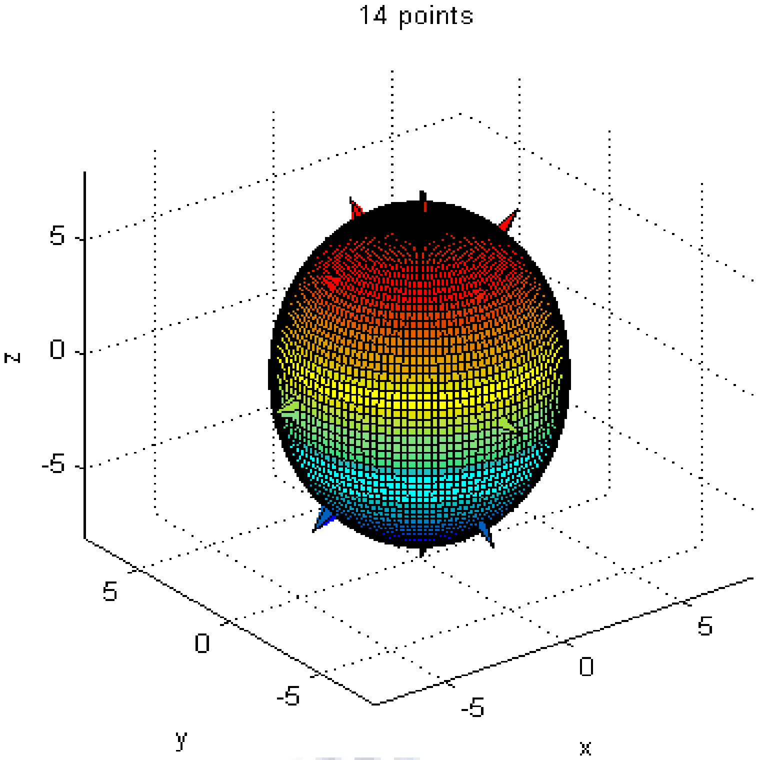


Figure 79: Pick these fourteen points which are symmetric and they are u(1,1), u(M/4,1), u(M/4,N/4), u(M/4, N/2), u(M/4, 3N/4), u(M/2,1), u(M/2, N/4), u(M/2, N/2), u(M/2, 3N/4), u(3M/4,1), u(3M/4, N/4), u(3M/4, N/2), u(3M/4, 3N/4), u(M,1).

Now, we are going to show the frequency analysis of the values in above fourteen points in $t = 1500{\sim}1600$ to implement that the spiral wave is periodic.

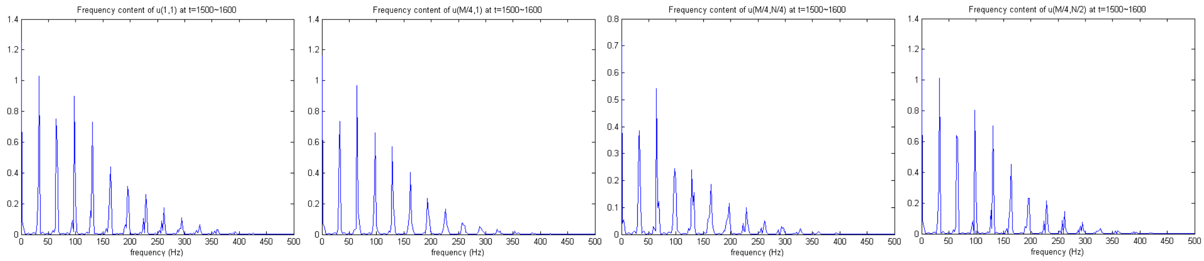


Figure 80: The frequency analysis of u at u(1,1), u(M/4,1),u(M/4,N/4), u(M/4, N/2).

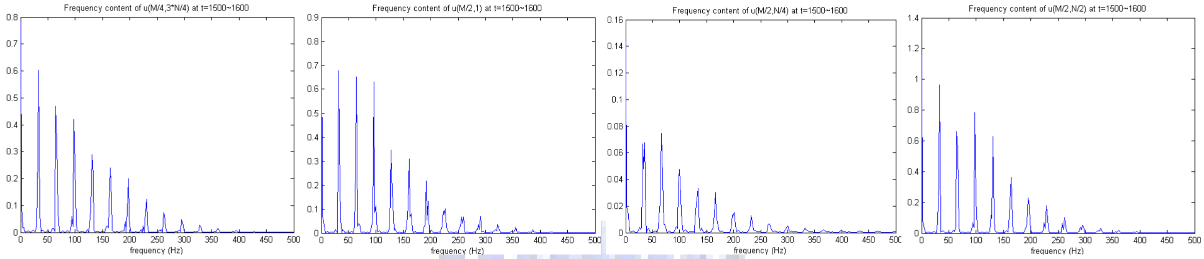


Figure 81: The frequency analysis of u at u(M/4, 3N/4), u(M/2,1), u(M/2, N/4), u(M/2, N/2).

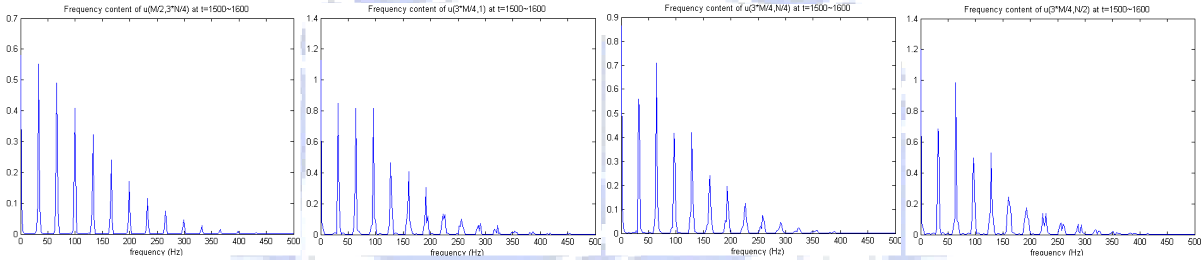


Figure 82: The frequency analysis of u at u(M/2, 3N/4), u(3M/4,1), u(3M/4, N/4), u(3M/4, N/2).

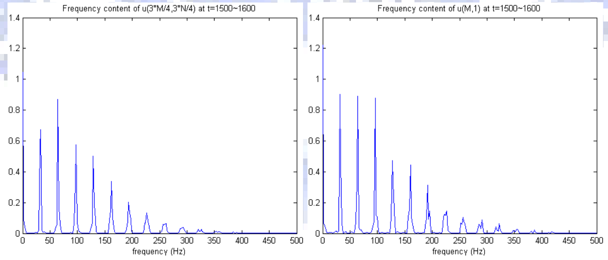


Figure 83: The frequency analysis of u at u(3M/4, 3N/4), u(M,1).

According to above Figure 80~ Figure 83, we could make sure that the spiral wave of the reaction-diffusion equation in our solver is periodic because they have the obvious property of periodicity. By the way, we find that the wave is getting more stable in a long time and the characteristic of periodicity is getting more obvious when we take smaller time step. In this section, we complete the numerical simulation of the reaction-diffusion equation with heart-like domain. The periodic results make us believe that the reaction-diffusion equation

(40) makes sense in physiology.

6、Explicit Numerical Solver of the Heat Equation

After solving the reaction-diffusion equation (40) in Chapter 5, we are interested in the motions of the convection-diffusion which domain keeps moving. Before we solve the convection-diffusion equation (We will introduce this in Chapter 7.), the first thing which we must do is to solve the heat equation in Curvilinear coordinates in this section.

6.1 Discretization of the Heat Equation

Heat equation can be described as

$$\Gamma_t = \Delta_s \Gamma = \nabla_s \cdot \nabla_s \Gamma \tag{45}$$

According to [7] and rearrange it by $\frac{\partial \Gamma}{\partial \phi} = \frac{\partial \Gamma}{\partial \mathbf{X}} \cdot \frac{\partial \mathbf{X}}{\partial \phi} = \nabla \Gamma \cdot \frac{\partial \mathbf{X}}{\partial \phi}$, we derive

$$\nabla_s \Gamma = \frac{\mathbf{b}_2 \frac{\partial \Gamma}{\partial \phi}\left|\frac{\partial \mathbf{X}}{\partial \theta}\right| + \mathbf{b}_1 \frac{\partial \Gamma}{\partial \theta}\left|\frac{\partial \mathbf{X}}{\partial \phi}\right|}{\left|\frac{\partial \mathbf{X}}{\partial \phi} \times \frac{\partial \mathbf{X}}{\partial \theta}\right|}$$

$$\nabla_s \cdot \nabla_s \Gamma = \frac{\mathbf{b}_2 \cdot \frac{\partial (\nabla_s \Gamma)}{\partial \phi}\left|\frac{\partial \mathbf{X}}{\partial \theta}\right| + \mathbf{b}_1 \cdot \frac{\partial (\nabla_s \Gamma)}{\partial \theta}\left|\frac{\partial \mathbf{X}}{\partial \phi}\right|}{\left|\frac{\partial \mathbf{X}}{\partial \phi} \times \frac{\partial \mathbf{X}}{\partial \theta}\right|}$$

where

$$\mathbf{b}_1 = \mathbf{n} \times \boldsymbol{\tau}_1 \quad , \quad \mathbf{b}_2 = \boldsymbol{\tau}_2 \times \mathbf{n}$$

$$\boldsymbol{\tau}_1 = \frac{\frac{\partial \mathbf{X}}{\partial \phi}}{\left|\frac{\partial \mathbf{X}}{\partial \phi}\right|}, \boldsymbol{\tau}_2 = \frac{\frac{\partial \mathbf{X}}{\partial \theta}}{\left|\frac{\partial \mathbf{X}}{\partial \theta}\right|}, \mathbf{n} = \frac{\frac{\partial \mathbf{X}}{\partial \phi} \times \frac{\partial \mathbf{X}}{\partial \theta}}{\left|\frac{\partial \mathbf{X}}{\partial \phi} \times \frac{\partial \mathbf{X}}{\partial \theta}\right|}$$

$\mathbf{X}$ means the spatial grids of the domain and $\mathbf{X}_{i,j} = \mathbf{X}(R, \phi_i, \theta_j, t) \approx \mathbf{X}(\phi_i, \theta_j . t)$ by setting r be a constant R and set

$$\mathbf{X}(R, \phi_i, \theta_j, t) = \left(R\sin(\phi_i)\cos(\theta_j), R\sin(\phi_i)\sin(\theta_j), R\cos(\phi_i), t\right)$$

Γ means the concentration on the grids of the domain and $\Gamma_{i,j} = \Gamma(R, \phi_i, \theta_j, t) \approx \Gamma(\phi_i, \theta_j, t)$ by setting r be a constant R and the grids we mesh here are as Figure 3. Solve equation (45) in Curvilinear coordinates and use explicit first-order Forward-Euler method,

$$\frac{\Gamma^{n+1}-\Gamma^{n}}{\Delta t}=\Delta_s\Gamma^{n}$$

i.e.

$$\Gamma^{n+1}=\Gamma^{n}+\Delta_s\Gamma^{n}\Delta t$$

Use the second-order central difference method for spatial and concentration variables:

$$\frac{\partial \mathbf{X}_{i,j}}{\partial \phi}\approx\frac{\mathbf{X}_{i+1,j}-\mathbf{X}_{i-1,j}}{2\Delta\phi}\quad,\quad\frac{\partial \mathbf{X}_{i,j}}{\partial \theta}\approx\frac{\mathbf{X}_{i,j+1}-\mathbf{X}_{i,j-1}}{2\Delta\theta}$$

$$\frac{\partial \Gamma_{i,j}}{\partial \phi}\approx\frac{\Gamma_{i+1,j}-\Gamma_{i-1,j}}{2\Delta\phi}\quad,\quad\frac{\partial \Gamma_{i,j}}{\partial \theta}\approx\frac{\Gamma_{i,j+1}-\Gamma_{i,j-1}}{2\Delta\theta}$$

The diagram of solving equation (45) is as follows.

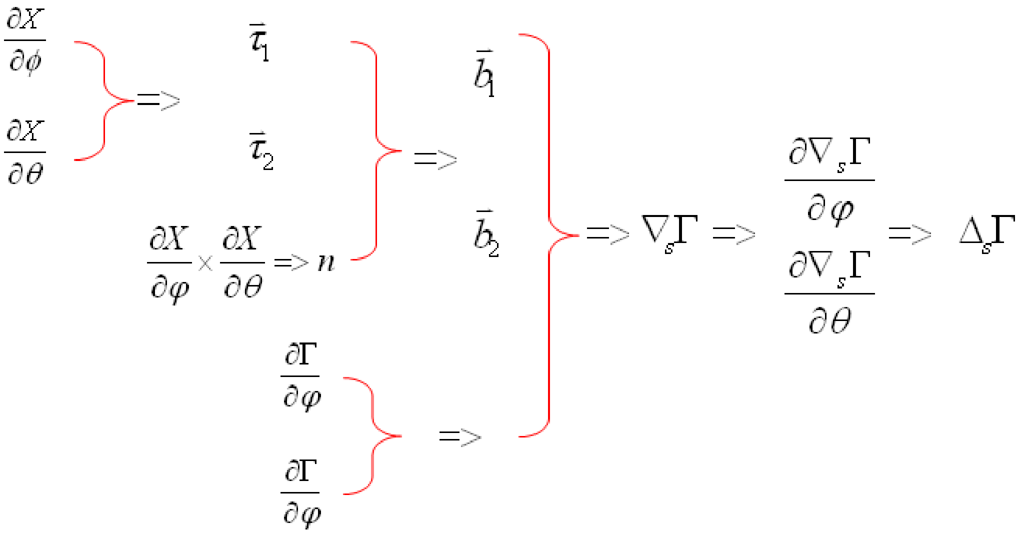

$\frac{\partial X}{\partial \phi}$
$\frac{\partial X}{\partial \theta}$
=>
$\vec{\tau}_1$
$\vec{\tau}_2$
$\frac{\partial X}{\partial \varphi}\times\frac{\partial X}{\partial \theta}$ => n
=>
$\vec{b}_1$
$\vec{b}_2$
=> $\nabla_s\Gamma$ =>
$\frac{\partial \nabla_s \Gamma}{\partial \varphi}$
$\frac{\partial \nabla_s \Gamma}{\partial \theta}$
=> $\Delta_s\Gamma$
$\frac{\partial \Gamma}{\partial \varphi}$
$\frac{\partial \Gamma}{\partial \varphi}$
=>

Figure 84: We must solve these values on each grid at each time step.

We want to construct $\Delta_s\Gamma$ at initial time step in our right hand side of (45) as above procedures. Given a clear notations and computational process as follows:

$$\frac{\partial \mathbf{X}_{i,j}}{\partial \phi}\approx\frac{\mathbf{X}_{i+1,j}-\mathbf{X}_{i-1,j}}{2\Delta\phi}\equiv\left(p_{1_{i,j}},p_{2_{i,j}},p_{3_{i,j}}\right)$$

$$\frac{\partial \mathbf{X}_{i,j}}{\partial \theta}\approx\frac{\mathbf{X}_{i,j+1}-\mathbf{X}_{i,j-1}}{2\Delta\theta}\equiv\left(q_{1_{i,j}},q_{2_{i,j}},q_{3_{i,j}}\right)$$

$$\left|\frac{\partial \mathbf{X}_{i,j}}{\partial \phi}\right|=\left({p_{1_{i,j}}}^{2}+{p_{2_{i,j}}}^{2}+{p_{3_{i,j}}}^{2}\right)^{\frac{1}{2}}=g_{i,j}$$

$$\left|\frac{\partial \mathbf{X}_{i,j}}{\partial \theta}\right|=\left({q_{1_{i,j}}}^{2}+{q_{2_{i,j}}}^{2}+{q_{3_{i,j}}}^{2}\right)^{\frac{1}{2}}=h_{i,j}$$

$$\boldsymbol{\tau}_{1_{i,j}} = \frac{(p_{1_{i,j}}, p_{2_{i,j}}, p_{3_{i,j}})}{\left({p_{1_{i,j}}}^2 + {p_{2_{i,j}}}^2 + {p_{3_{i,j}}}^2\right)^{\frac{1}{2}}} \equiv \left(\boldsymbol{\tau}_{1_{i,j_1}}, \boldsymbol{\tau}_{1_{i,j_2}}, \boldsymbol{\tau}_{1_{i,j_3}}\right)$$

$$\boldsymbol{\tau}_{2_{i,j}} = \frac{(q_{1_{i,j}}, q_{2_{i,j}}, q_{3_{i,j}})}{\left({q_{1_{i,j}}}^2 + {q_{2_{i,j}}}^2 + {q_{3_{i,j}}}^2\right)^{\frac{1}{2}}} \equiv \left(\boldsymbol{\tau}_{2_{i,j_1}}, \boldsymbol{\tau}_{2_{i,j_2}}, \boldsymbol{\tau}_{2_{i,j_3}}\right)$$

$$\frac{\partial \mathbf{X}_{i,j}}{\partial \phi} \times \frac{\partial \mathbf{X}_{i,j}}{\partial \theta} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ p_{1_{i,j}} & p_{2_{i,j}} & p_{3_{i,j}} \\ q_{1_{i,j}} & q_{2_{i,j}} & q_{3_{i,j}} \end{vmatrix}$$

$$= \left(p_{2_{i,j}} q_{3_{i,j}} - q_{2_{i,j}} p_{3_{i,j}}, p_{3_{i,j}} q_{1_{i,j}} - q_{3_{i,j}} p_{1_{i,j}}, p_{1_{i,j}} q_{2_{i,j}} - q_{1_{i,j}} p_{2_{i,j}}\right)$$

$$\equiv \left(c_{i,j_1}, c_{i,j_2}, c_{i,j_3}\right)$$

$$\left|\frac{\partial \mathbf{X}_{i,j}}{\partial \phi} \times \frac{\partial \mathbf{X}_{i,j}}{\partial \theta}\right| = \left({c_{i,j_1}}^2 + {c_{i,j_1}}^2 + {c_{i,j_3}}^2\right)^{\frac{1}{2}} = d_{i,j}$$

$$\mathbf{n}_{i,j} = \frac{\left(c_{i,j_1}, c_{i,j_1}, c_{i,j_3}\right)}{\left({c_{i,j_1}}^2 + {c_{i,j_1}}^2 + {c_{i,j_3}}^2\right)^{\frac{1}{2}}} \equiv (\mathbf{n}_{i,j_\mathbf{1}}, \mathbf{n}_{i,j_\mathbf{2}}, \mathbf{n}_{i,j_\mathbf{3}})$$

$$\mathbf{b}_{1_{i,j}} = \mathbf{n}_{i,j} \times \boldsymbol{\tau}_{1_{i,j}} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ \mathbf{n}_{i,j_\mathbf{1}} & \mathbf{n}_{i,j_\mathbf{2}} & \mathbf{n}_{i,j_\mathbf{3}} \\ \boldsymbol{\tau}_{1_{i,j_1}} & \boldsymbol{\tau}_{1_{i,j_2}} & \boldsymbol{\tau}_{1_{i,j_3}} \end{vmatrix}$$

$$= \left(\mathbf{n}_{i,j_\mathbf{2}} \boldsymbol{\tau}_{1_{i,j_3}} - \boldsymbol{\tau}_{1_{i,j_2}} \mathbf{n}_{i,j_\mathbf{3}}, \mathbf{n}_{i,j_\mathbf{3}} \boldsymbol{\tau}_{1_{i,j_1}} - \boldsymbol{\tau}_{1_{i,j_3}} \mathbf{n}_{i,j_\mathbf{1}}, \mathbf{n}_{i,j_\mathbf{1}} \boldsymbol{\tau}_{1_{i,j_2}} - \boldsymbol{\tau}_{1_{i,j_1}} \mathbf{n}_{i,j_\mathbf{2}}\right)$$

$$\equiv \left(\mathbf{b}_{1_{i,j_1}}, \mathbf{b}_{1_{i,j_2}}, \mathbf{b}_{1_{i,j_3}}\right)$$

$$\mathbf{b}_{2_{i,j}} = \boldsymbol{\tau}_{2_{i,j}} \times \mathbf{n}_{i,j} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ \boldsymbol{\tau}_{2_{i,j_1}} & \boldsymbol{\tau}_{2_{i,j_2}} & \boldsymbol{\tau}_{2_{i,j_3}} \\ \mathbf{n}_{i,j_\mathbf{1}} & \mathbf{n}_{i,j_\mathbf{2}} & \mathbf{n}_{i,j_\mathbf{3}} \end{vmatrix}$$

$$= \left(\boldsymbol{\tau}_{2_{i,j_2}} \mathbf{n}_{i,j_\mathbf{3}} - \mathbf{n}_{i,j_\mathbf{2}} \boldsymbol{\tau}_{2_{i,j_3}}, \boldsymbol{\tau}_{2_{i,j_3}} \mathbf{n}_{i,j_\mathbf{1}} - \mathbf{n}_{i,j_\mathbf{3}} \boldsymbol{\tau}_{2_{i,j_1}}, \boldsymbol{\tau}_{2_{i,j_1}} \mathbf{n}_{i,j_\mathbf{2}} - \mathbf{n}_{i,j_\mathbf{1}} \boldsymbol{\tau}_{2_{i,j_2}}\right)$$

$$\equiv \left(\mathbf{b}_{2_{i,j_1}}, \mathbf{b}_{2_{i,j_2}}, \mathbf{b}_{2_{i,j_3}}\right)$$

$$\frac{\partial \Gamma_{i,j}}{\partial \phi} \approx \frac{\Gamma_{i+1,j} - \Gamma_{i-1,j}}{2\Delta\phi} \equiv \Gamma_{\phi_{i,j}}$$

$$\frac{\partial \Gamma_{i,j}}{\partial \theta} \approx \frac{\Gamma_{i,j+1} - \Gamma_{i,j-1}}{2\Delta\theta} \equiv \Gamma_{\theta_{i,j}}$$

According to above values, we could compute

$$\nabla_s\Gamma_{i,j} = \frac{\left(\mathbf{b}_{2_{i,j_1}}, \mathbf{b}_{2_{i,j_2}}, \mathbf{b}_{2_{i,j_3}}\right)\Gamma_{\phi_{i,j}} h_{i,j} + \left(\mathbf{b}_{1_{i,j_1}}, \mathbf{b}_{1_{i,j_2}}, \mathbf{b}_{1_{i,j_3}}\right)\Gamma_{\theta_{i,j}} g_{i,j}}{d_{i,j}}$$

$$\frac{\partial \nabla_s\Gamma_{i,j}}{\partial\phi} \approx \frac{\nabla_s\Gamma_{i+1,j} - \nabla_s\Gamma_{i-1,j}}{2\Delta\phi} \equiv \left(\tilde{\Gamma}_{\phi_{i,j_1}}, \tilde{\Gamma}_{\phi_{i,j_2}}, \tilde{\Gamma}_{\phi_{i,j_3}}\right)$$

$$\frac{\partial \nabla_s\Gamma_{i,j}}{\partial\theta} \approx \frac{\nabla_s\Gamma_{i,j+1} - \nabla_s\Gamma_{i,j-1}}{2\Delta\theta} \equiv \left(\tilde{\Gamma}_{\theta_{i,j_1}}, \tilde{\Gamma}_{\theta_{i,j_2}}, \tilde{\Gamma}_{\theta_{i,j_3}}\right)$$

$$\Delta_s\Gamma_{i,j} = \frac{\left(\mathbf{b}_{2_{i,j_1}}, \mathbf{b}_{2_{i,j_2}}, \mathbf{b}_{2_{i,j_3}}\right)\cdot\left(\tilde{\Gamma}_{\phi_{i,j_1}}, \tilde{\Gamma}_{\phi_{i,j_2}}, \tilde{\Gamma}_{\phi_{i,j_3}}\right) h_{i,j} + \left(\mathbf{b}_{1_{i,j_1}}, \mathbf{b}_{1_{i,j_2}}, \mathbf{b}_{1_{i,j_3}}\right)\cdot\left(\tilde{\Gamma}_{\theta_{i,j_1}}, \tilde{\Gamma}_{\theta_{i,j_2}}, \tilde{\Gamma}_{\theta_{i,j_3}}\right) g_{i,j}}{d_{i,j}}$$

Eventually, explicit numerical heat solver will arise from above discretization. But we still need the symmetric properties of the spherical surface domain. In other words, these are the excellences in geometry. We need to know $\mathbf{X}_{0,j}$ and $\mathbf{X}_{M+1,j}$ to compute $\frac{\partial \mathbf{X}_{i,j}}{\partial\phi}$ as follows (Γ use the same properties),

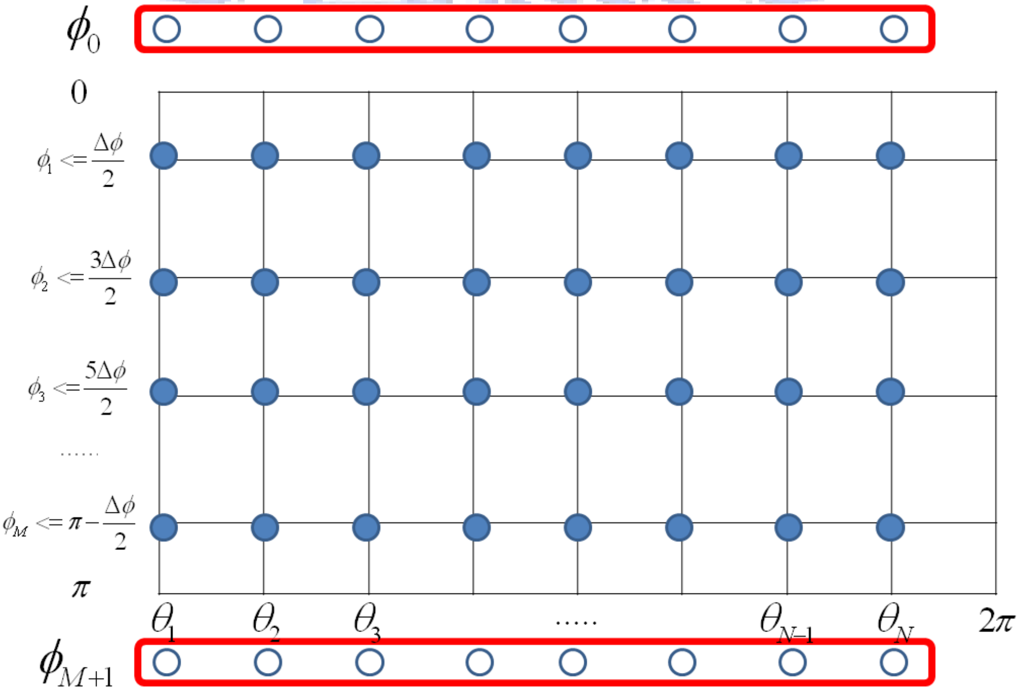

ϕ_0
0
$\phi_1 <= \frac{\Delta\phi}{2}$
$\phi_2 <= \frac{3\Delta\phi}{2}$
$\phi_3 <= \frac{5\Delta\phi}{2}$
……
$\phi_M <= \pi - \frac{\Delta\phi}{2}$
π
θ_1
θ_2
θ_3
…..
θ_{N-1}
θ_N
2π
ϕ_{M+1}

Figure 85: The points we circled are unknowns.

Solve them by symmetric property on the spherical surface domain as follows,

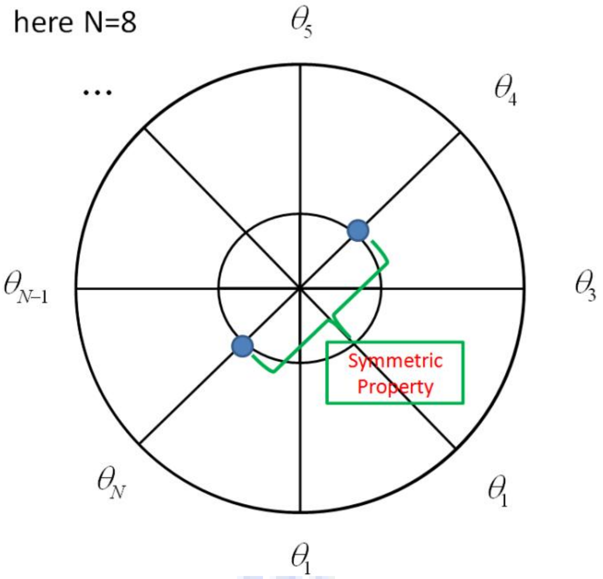


Figure 86: Symmetric properties.

$$\begin{cases} \mathbf{X}_{0,j} = \mathbf{X}_{1,j+\frac{N}{2}} \ , \ \text{if } j \le \frac{N}{2} \\ \mathbf{X}_{0,j} = \mathbf{X}_{1,j-\frac{N}{2}} \ , \ \text{if } j > \frac{N}{2} \end{cases}$$

$$\begin{cases} \mathbf{X}_{M+1,j} = \mathbf{X}_{M,j+\frac{N}{2}} \ , \ \text{if } j \le \frac{N}{2} \\ \mathbf{X}_{M+1,j} = \mathbf{X}_{M,j-\frac{N}{2}} \ , \ \text{if } j > \frac{N}{2} \end{cases}$$

We need to know $\mathbf{X}_{i,0}$ and $\mathbf{X}_{i,N+1}$ to compute $\frac{\partial \mathbf{X}_{i,j}}{\partial \theta}$ as follows (Γ use the same properties.),

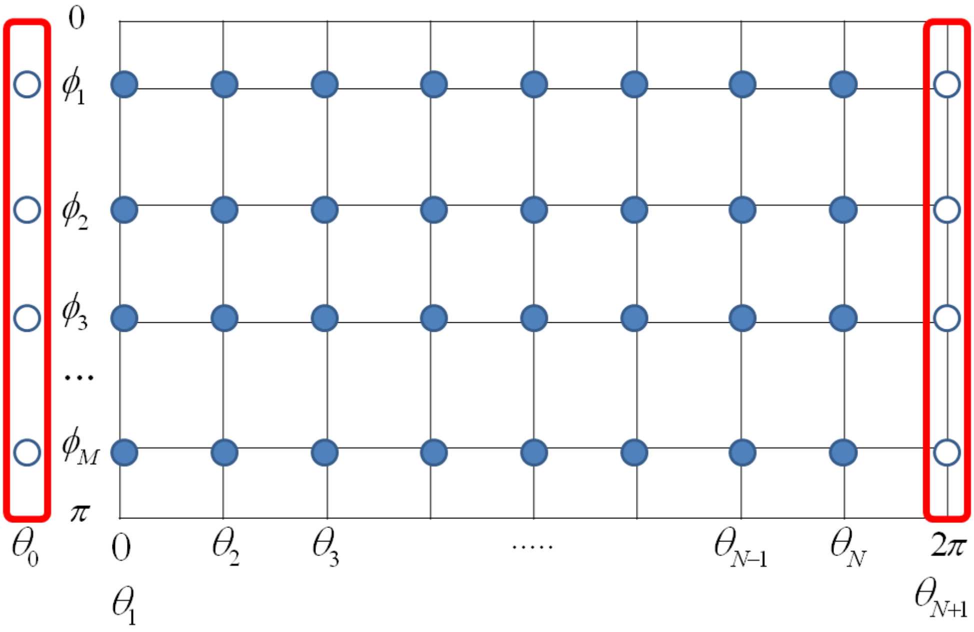


Figure 87: The points we circled are unknowns.

Solve them by below simple properties on our spherical domain,

$$\mathbf{X}(\mathrm{R},\phi_{\mathrm{i}},0,\mathrm{t})=\mathbf{X}(\mathrm{R},\phi_{\mathrm{i}},2\pi,\mathrm{t})$$

So

$$\mathbf{X}(\mathrm{R},\phi_{\mathrm{i}},\theta_{1},\mathrm{t})=\mathbf{X}(\mathrm{R},\phi_{\mathrm{i}},\theta_{\mathrm{N}+1},\mathrm{t})$$

After imposing above two kinds of properties on the spherical surface domain, it is easy to get the values at next time step of $\Gamma^{\mathrm{n}+1}$ by the old values Γ^{n} and $\Delta_{\mathrm{s}}\Gamma^{\mathrm{n}}$. The results will be shown in Section 6.2.1. According to the result of Chapter 3, symmetric discretization for spatial terms will perform good outcome of mass conservation. But solve equation (45) by the form of discretization as Section 6.1, we have no way of using this method. Seek what is less attractive than our original objective. We use the discretization as (9) which is like the spirit of symmetric discretization for ϕ and θ, and we hope to see something good in total mass. We will show two solvers with below the methods in Section 6.2.1 and Section 6.2.2, respectively and do nothing change in Γ.

Method 1:

Use the second-order central difference method 1 below for spatial variables as Section 6.1:

$$\frac{\partial\mathbf{X}_{\mathrm{i,j}}}{\partial\phi}\approx\frac{\mathbf{X}_{\mathrm{i+1,j}}-\mathbf{X}_{\mathrm{i-1,j}}}{2\Delta\phi}\quad,\quad\frac{\partial\mathbf{X}_{\mathrm{i,j}}}{\partial\theta}\approx\frac{\mathbf{X}_{\mathrm{i,j+1}}-\mathbf{X}_{\mathrm{i,j-1}}}{2\Delta\theta}$$

The results will be shown in Section 6.3.1.

Method 2:

Use the second-order central difference method 2 below for spatial variables cover for that in Section 6.1:

$$\frac{\partial\mathbf{X}_{\mathrm{i,j}}}{\partial\phi}\approx\frac{\mathbf{X}_{\mathrm{i}+\frac{1}{2},\mathrm{j}}-\mathbf{X}_{\mathrm{i}-\frac{1}{2},\mathrm{j}}}{\Delta\phi},\quad\frac{\partial\mathbf{X}_{\mathrm{i,j}}}{\partial\theta}\approx\frac{\mathbf{X}_{\mathrm{i,j}+\frac{1}{2}}-\mathbf{X}_{\mathrm{i,j}-\frac{1}{2}}}{\Delta\theta}\quad\forall\mathrm{i},\mathrm{for}\ \mathrm{j}=1,\ldots,\mathrm{N}.$$

Impose below techniques on (45),

$$\begin{cases}\mathbf{X}_{\mathrm{i}+\frac{1}{2},\mathrm{j}}=\dfrac{\mathbf{X}_{\mathrm{i,j}}+\mathbf{X}_{\mathrm{i+1,j}}}{2}\\ \mathbf{X}_{\mathrm{i}-\frac{1}{2},\mathrm{j}}=\dfrac{\mathbf{X}_{\mathrm{i-1,j}}+\mathbf{X}_{\mathrm{i,j}}}{2}\end{cases}$$

$$\begin{cases} \mathbf{X}_{i,j+\frac{1}{2}} = \dfrac{\mathbf{X}_{i,j} + \mathbf{X}_{i,j+1}}{2} \\ \mathbf{X}_{i,j-\frac{1}{2}} = \dfrac{\mathbf{X}_{i,j-1} + \mathbf{X}_{i,j}}{2} \end{cases}$$

The differences of our schemes will exist only in spatial discretization with method 2. The results will be shown in Section 6.3.2.

6.2 Results of the Explicit Numerical Solvers for the Heat Equation

6.2.1 Test Accuracy of the Solvers

After producing the solvers by Section 6.1, the most important thing for us is to make sure that our solvers are correct. Here impose the exact solution as

$$\Gamma = e^{-2\frac{t}{R^2}}(x + y + z) = e^{-2\frac{t}{R^2}}(R\sin(\phi)\cos(\theta) + R\sin(\phi)\sin(\theta) + R\cos(\phi))$$

And R is a constant. Because we administer explicit Forward-Euler method in temporal term, we must choose the smaller Δt and it leads to spend more time to get the numerical solutions. According to CFL condition, the stability of the numerical method in equation (45) with this numerical method (Forward-Euler) needs to satisfy the property

$$\Delta t = O(\Delta x^2)$$

In our programming, we could pick out

$$\Delta t = \frac{1}{2(2\pi)^2}\Delta x^2 = \frac{1}{2}\frac{1}{N^2}$$

Here provide the exact solution at $T = 1$ for (45) as follows,

$$u = e^{-2\frac{1}{R^2}}(R\sin(\phi)\cos(\theta) + R\sin(\phi)\sin(\theta) + R\cos(\phi))$$

We explicitly solved the heat equation (45) $\Gamma_t = \Delta_s\Gamma$ on the spherical surface domain by Section 6.1 with central difference method 1. In table 9, solve (45) numerically with $R = 5$ at $T = 1$ and check it by the exact solution

$$u = e^{-\frac{2}{25}}(5\sin(\phi)\cos(\theta) + 5\sin(\phi)\sin(\theta) + 5\cos(\phi))$$

$M \times N$	L_∞nrom	RATIO	ORDER
8×16	0.0083	0	0
16×32	0.0021	3.9524	1.9827
32×64	5.3821e-004	3.9018	1.9641
64×128	1.3924e-004	3.8653	1.9506

Table 9: The results of the numerical accuracy test.

We explicitly solved the heat equation (45) $\Gamma_t = \Delta_s \Gamma$ on the spherical surface domain by Section 6.1 but replaced central difference method 1 by central difference method 2 in table 2.

$M \times N$	L_∞nrom	RATIO	ORDER
8×16	0.0127	0	0
16×32	0.0033	3.8485	1.9443
32×64	8.3253e-004	3.9638	1.9869
64×128	2.1200e-004	3.9270	1.9734

Table 10: The results of the numerical accuracy test.

The results show that we got correct solvers.

6.3 Mass Conservation

We developed an interest in doing justice to the variation of total mass in our numerical solver due to

$$\frac{\partial \Gamma}{\partial t} = \Delta_s \Gamma = \nabla_s^2 \Gamma$$

Integrate both sides,

$$\iint_\Omega \frac{\partial \Gamma}{\partial t} dV = \iint_\Omega \nabla_s^2 \Gamma dV \qquad (46)$$

From [16],

$$= \iint\limits_{\Omega} \nabla^2\Gamma - \frac{\partial^2\Gamma}{\partial n^2} - \frac{\partial\Gamma}{\partial n} dV$$

We impose our assumption

$$\frac{\partial\Gamma}{\partial n} = 0$$

It will lead to

$$\frac{\partial^2\Gamma}{\partial n^2} = 0$$

So, (46) becomes

$$\iint\limits_{\Omega} \frac{\partial\Gamma}{\partial t} dV = \iint\limits_{\Omega} \nabla^2\Gamma dV$$

$$= \iint\limits_{\Omega} \nabla \cdot (\nabla\Gamma) dV$$

According to the Divergence Theorem as follows,

$$\int\limits_{\Omega} \nabla \cdot \Gamma d\mathbf{X} = \int\limits_{\partial\Omega} \Gamma \cdot n d\mathbf{A}$$

So, (46) becomes

$$\iint\limits_{\Omega} \frac{\partial\Gamma}{\partial t} dV = \iint\limits_{\partial\Omega} \nabla\Gamma \cdot n dS = \iint\limits_{\partial\Omega} n \cdot \nabla\Gamma dS$$

$$= \iint\limits_{\partial\Omega} \frac{\partial\Gamma}{\partial n} dS$$

$$= 0$$

Here, we have the result

$$\iint\limits_{\Omega} \frac{\partial\Gamma}{\partial t} dV = 0$$

Transform coordinate in integration and $V = S$, $\Omega = \partial\Omega$ in our domain

$$\iint\limits_{\partial\Omega}\frac{\partial\Gamma}{\partial t}dS=\frac{\partial}{\partial t}\iint\limits_{\partial\Omega}\Gamma dS=\frac{\partial}{\partial t}\iint\limits_{\partial\Omega}\Gamma dxdydz=\frac{\partial}{\partial t}\int_0^{2\pi}\int_0^{\pi}\Gamma\left|\frac{\partial\mathbf{X}}{\partial\phi}\times\frac{\partial\mathbf{X}}{\partial\theta}\right|d\phi d\theta$$

Total mass is

$$\int_0^{2\pi}\int_0^{\pi}\Gamma\left|\frac{\partial\mathbf{X}}{\partial\phi}\times\frac{\partial\mathbf{X}}{\partial\theta}\right|d\phi d\theta$$

After discretizing, it turns to be

$$\sum_{\substack{i=1,\dots,M\\ j=1,\dots,N}}u_{i,j}\left|\frac{\partial\mathbf{X}_{i,j}}{\partial\phi}\times\frac{\partial\mathbf{X}_{i,j}}{\partial\theta}\right|\Delta\phi\Delta\theta$$

We computed the total mass as above. But only show the last total mass at $T = 5$ that is not like Section 3 and Section 4 which compute till $T = 5$. Because it will cost us too much time.

6.3.1 The Mass of the Solvers with Section 6.1 and Central Difference Method 1

Use the solver of the heat equation (45) $\Gamma_t = \Delta_s\Gamma$ on the spherical surface domain by Section 6.1 with central difference method 1. Initial settings: $M = 32, N = 64, R = 5$, $\Delta t = \frac{1}{2}\frac{1}{(2\pi)^2}\Delta x^2 = \frac{1}{2}\frac{1}{N^2}$. Domain is the same as Figure 7.

Case 1: Initial setting is the same as Figure 8,

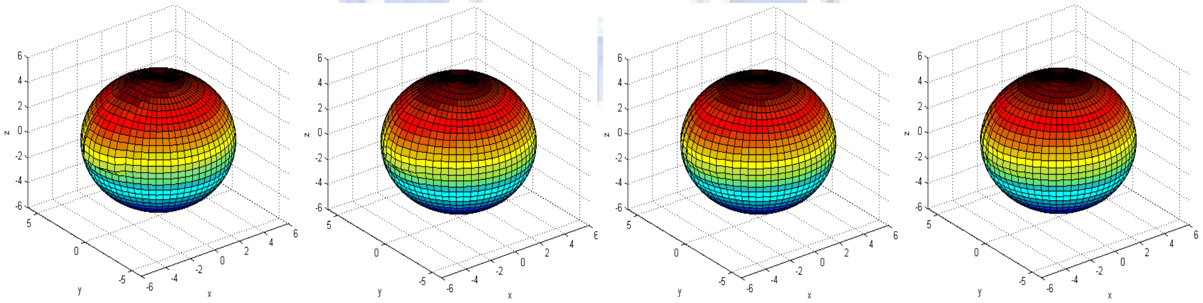


Figure 88: Left, $T = 0.25$, mass $= 14.7325$, relative error $= 2.4338e-005$.
Mid-left, $T = 0.5$, mass $= 14.7325$, relative error $= 2.4076e-005$.
Mid-right, $T = 0.75$, mass $= 14.7325$, relative error $= 2.3819e-005$.
Right, $T = 0.1$, mass $= 14.7325$, relative error $= 2.3562e-005$.

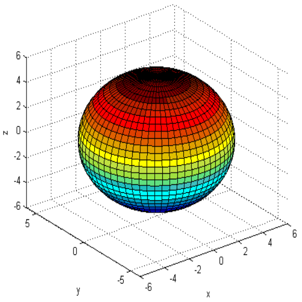


Figure 89： $\mathrm{T} = 5$, $\mathrm{mass} = 14.7324$, $\text{relative error} = 1.8856\mathrm{e}-005$.

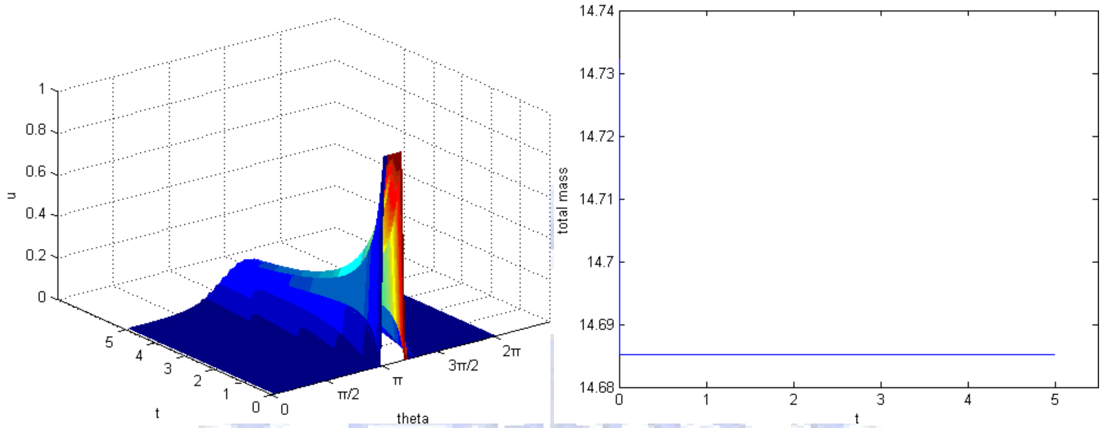


Figure 90： Left, the values of u on the equator change with time.
Right, total mass of u changes with time in $0\sim5$ seconds.

According to Figure 88, Figure 89, Figure 90, u was diffused as time goes by to be like the look of the original domain. Total mass is getting decreasing a little bit, but it will turn to stable that means mass will not change any more in a long time. Case 1 does not comply with the mass conservation law.

Case 2: Initial setting is the same as Figure 12,

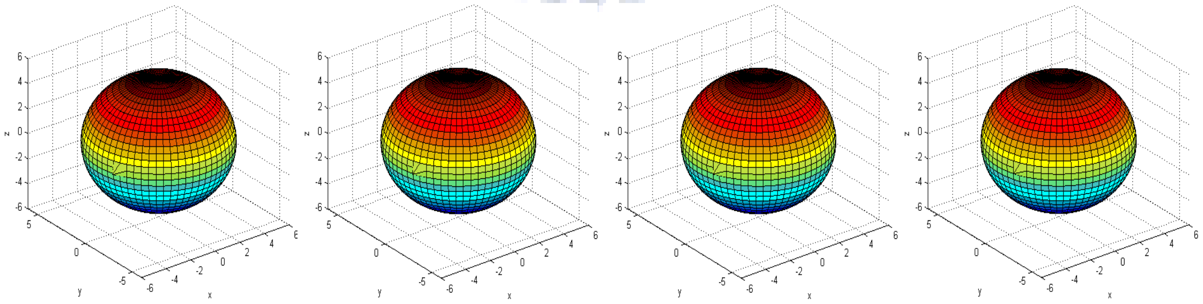


Figure 91： Left, $\mathrm{T} = 1/64$, $\mathrm{mass} = 0.24067$, $\text{relative error} = 4.2098\mathrm{e}-006$.
Mid-left, $\mathrm{T} = 1/32$, $\mathrm{mass} = 0.24067$, $\text{relative error} = 7.2202\mathrm{e}-006$.
Mid-right, $\mathrm{T} = 3/64$, $\mathrm{mass} = 0.24067$, $\text{relative error} = 1.0231\mathrm{e}-005$.
Right, $\mathrm{T} = 1/16$, $\mathrm{mass} = 0.24067$, $\text{relative error} = 1.3241\mathrm{e}-005$.

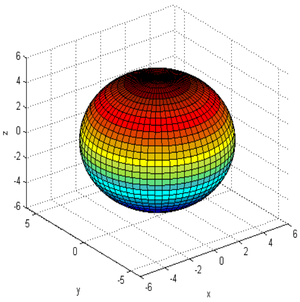


Figure 92： $T = 5$, mass $= 0.24097$, relative error $= 0.0012786$.

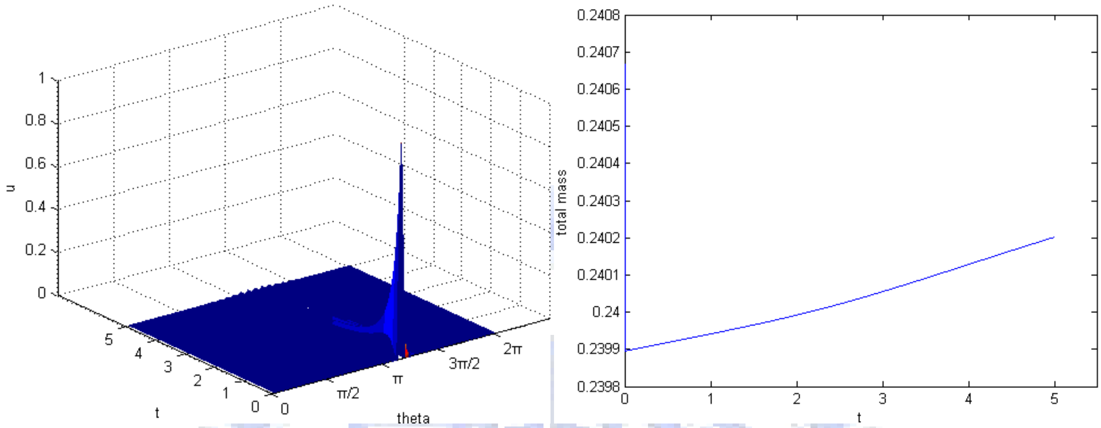


Figure 93： Left, the values of u on the equator change with time.
Right, total mass of u changes with time in $0 \sim 5$ seconds.

According to Figure 91, Figure 92, Figure 93, u was diffused as time goes by to be like the look of the original domain. Total mass is getting increasing a little bit, but it will turn to stable that means mass will not change any more in a long time. Case 2 does not comply with the mass conservation law.

Case 3: Initial setting is the same as Figure 16,

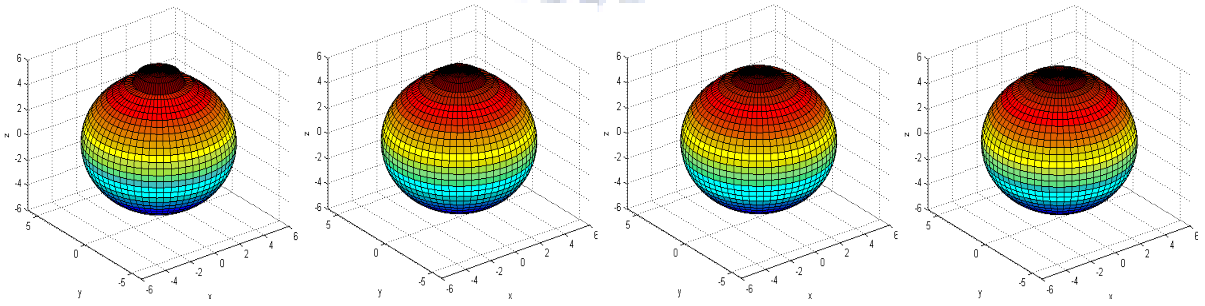


Figure 94： Left, $T = 0.25$, mass $= 6.7316$, relative error $= 0.0051562$.
Mid-left, $T = 0.5$, mass $= 6.6839$, relative error $= 0.012208$.
Mid-right, $T = 0.75$, mass $= 6.6464$, relative error $= 0.017755$.
Right, $T = 1$, mass $= 6.6195$, relative error $= 0.021727$.

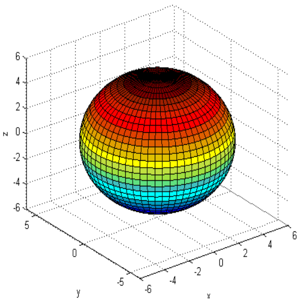


Figure 95: $\mathrm{T} = 5$, $\mathrm{mass} = 6.526$, $\mathrm{relative\ error} = 0.035539$.

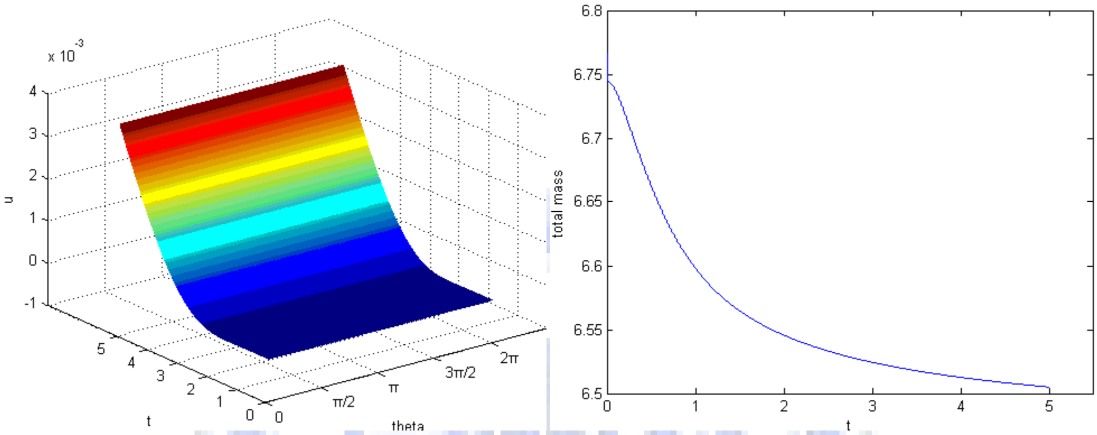


Figure 96: Left, the values of u on the equator change with time.
Right, total mass of u changes with time in $0{\sim}5$ seconds.

According to Figure 94, Figure 95, Figure 96, u was diffused as time goes by to be like the look of the original domain. Total mass is getting decreasing, but it will turn to stable that means mass will not change any more in a long time. Case 3 does not comply with the mass conservation law.

Case 4: Initial setting is the same as Figure 20,

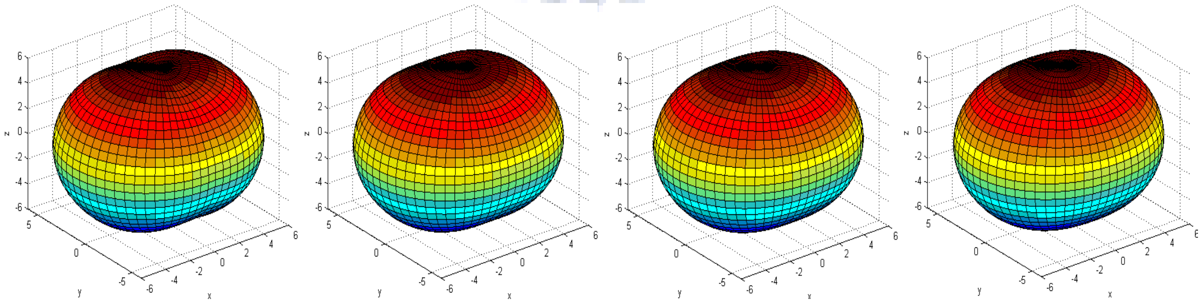


Figure 97: Left, $\mathrm{T} = 0.25$, $\mathrm{mass} = 446.7228$, $\mathrm{relative\ error} = 0.00014121$.
Mid-left, $\mathrm{T} = 0.5$, $\mathrm{mass} = 446.7606$, $\mathrm{relative\ error} = 0.0002258$.
Mid-right, $\mathrm{T} = 0.75$, $\mathrm{mass} = 446.7887$, $\mathrm{relative\ error} = 0.00028867$.
Right, $\mathrm{T} = 1$, $\mathrm{mass} = 446.8119$, $\mathrm{relative\ error} = 0.00034063$.

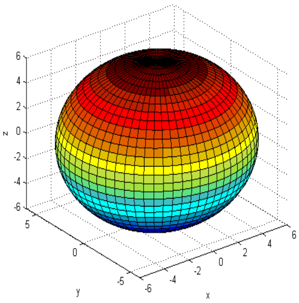


Figure 98: $T = 5$, $\text{mass} = 446.9892$, $\text{relative error} = 0.00073756$.

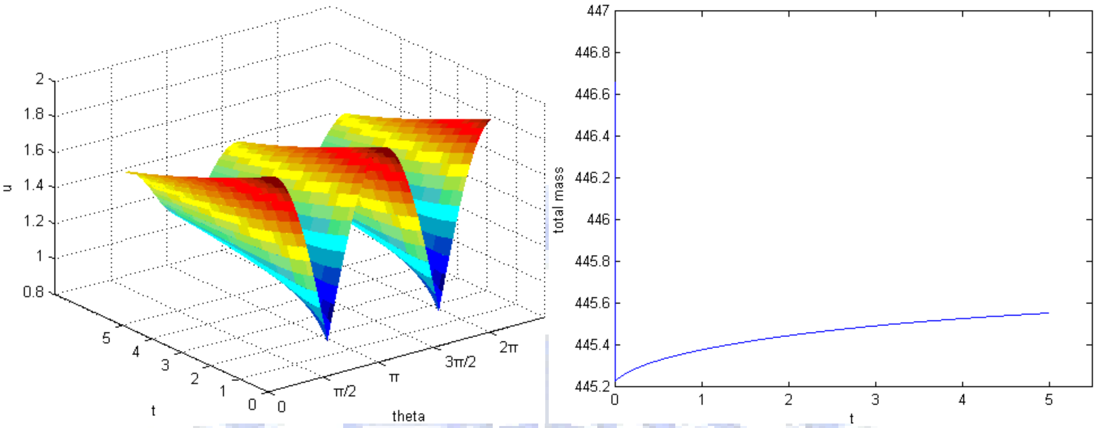


Figure 99: Left, the values of u on the equator change with time.
Right, total mass of u changes with time in $0\sim5$ seconds.

According to Figure 97, Figure 98, Figure 99, u was diffused as time goes by to be like the look of the original domain. Total mass is getting increasing, but it will turn to stable that means mass will not change any more in a long time. Case 4 does not comply with the mass conservation law.

In above Case 1~Case 4, we derive that Case 1 can get the better result by mass conservation with some machine error; the others show up a little bit derivation in our relative error.

6.3.2 The Mass of the Solver with Section 6.1 and Central Difference Method 2

Use the solver of the heat equation (45) $\Gamma_t = \Delta_s\Gamma$ on the spherical surface domain by Section 6.1 with central difference method 2. Initial settings: $M = 32, N = 64$, $R = 5$, $\Delta t = \frac{1}{2}\frac{1}{(2\pi)^2}\Delta x^2 = \frac{1}{2}\frac{1}{N^2}$. Domain is the same as Figure 7.

Case 1: Initial setting is the same as Figure 8,

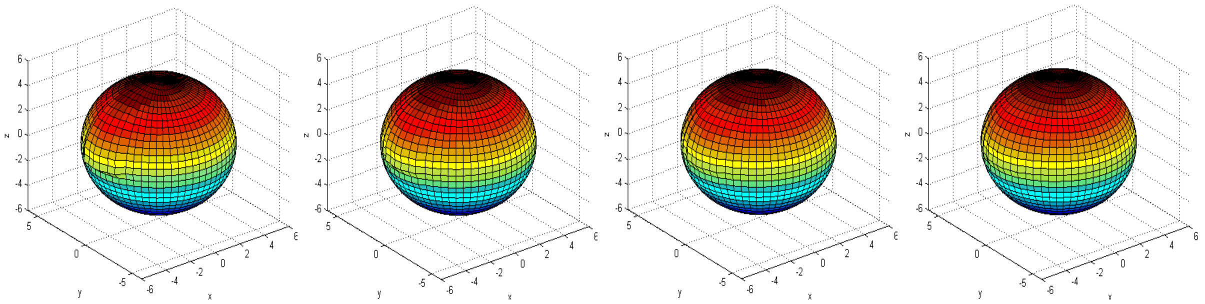


Figure 100：Left, $T = 0.25$, $mass = 14.7325$, $relative\ error = 2.428e - 005$.
Mid-left, $T = 0.5$, $mass = 14.7325$, $relative\ error = 2.4019e - 005$.
Mid-right, $T = 0.75$, $mass = 14.7325$, $relative\ error = 2.3764e - 005$.
Right, $T = 0.1$, $mass = 14.7325$, $relative\ error = 2.3508e - 005$.

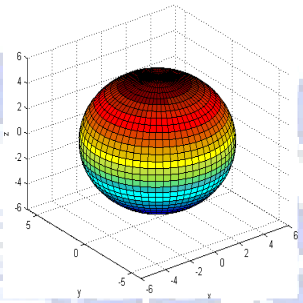


Figure 101：$T = 5$, $mass = 14.7324$, $relative\ error = 1.8826e - 005$.

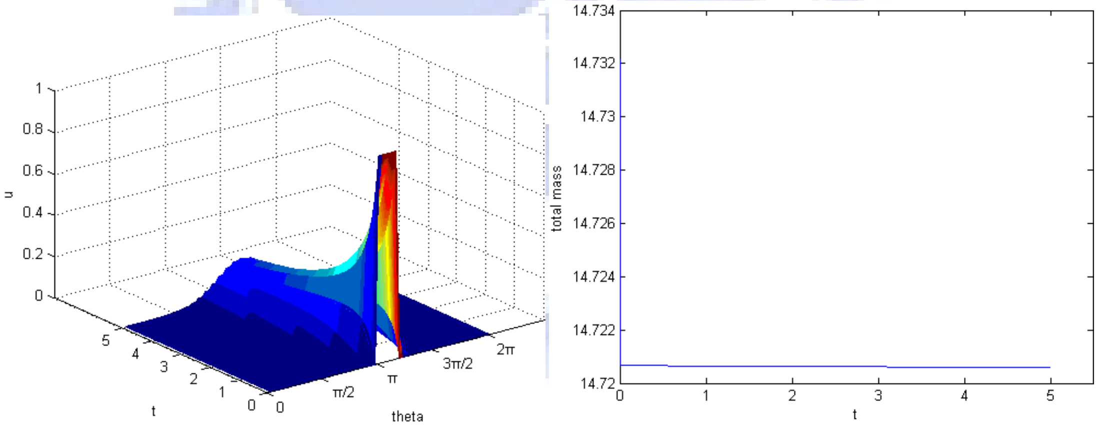


Figure 102：Left, the values of u on the equator change with time.
Right, total mass of u changes with time in $0 \sim 5$ seconds.

According to Figure 100, Figure 101, Figure 102, u was diffused as time goes by to be like the look of the original domain. Total mass is getting decreasing a little bit, but it will turn to stable that means mass will not change any more in a long time. Case 1 does not comply with the mass conservation law.

Case 2: Initial setting is the same as Figure 12,

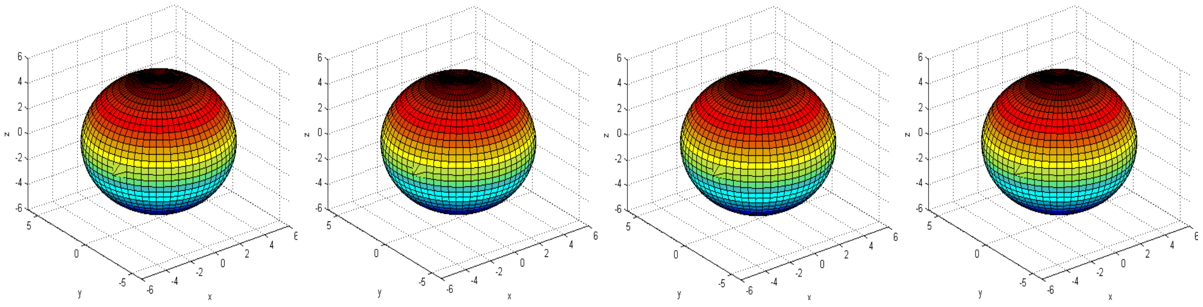


Figure 103: Left, $\mathrm{T} = 1/64$, $\text{mass} = 0.24067$, $\text{relative error} = 4.1997\mathrm{e} - 006$.

Mid-left, $\mathrm{T} = 1/32$, $\text{mass} = 0.24067$, $\text{relative error} = 7.2028\mathrm{e} - 006$.

Mid-right, $\mathrm{T} = 3/64$, $\text{mass} = 0.24067$, $\text{relative error} = 1.0206\mathrm{e} - 005$.

Right, $\mathrm{T} = 1/16$, $\text{mass} = 0.24067$, $\text{relative error} = 1.3209\mathrm{e} - 005$.

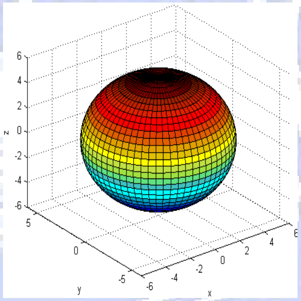


Figure 104: $\mathrm{T} = 5$, $\text{mass} = 0.24097$, $\text{relative error} = 0.001275$.

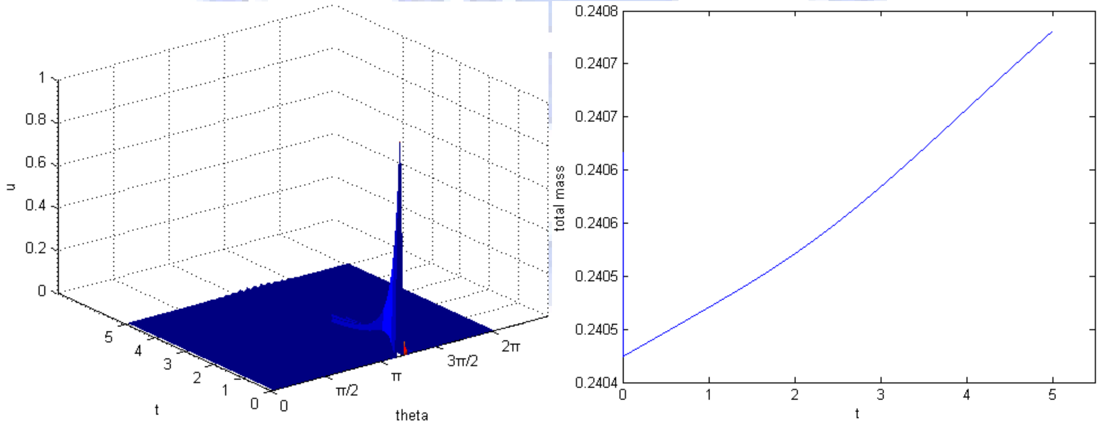


Figure 105: Left, the values of u on the equator change with time.

Right, total mass of u changes with time in 0~5 seconds.

According to Figure 103, Figure 104, Figure 105, u was diffused as time goes by to be like the look of the original domain. Total mass is getting increasing a little bit, but it will turn to stable that means mass will not change any more in a long time. Case 2 does not comply with the mass conservation law.

Case 3: Initial setting is the same as Figure 16,

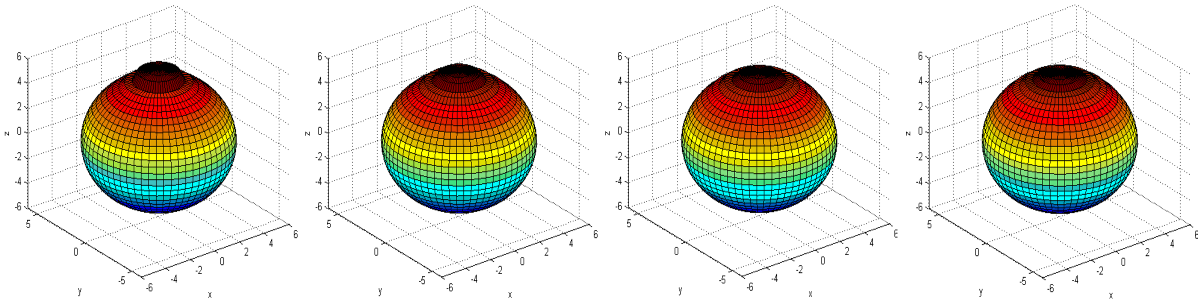


Figure 106: Left, $T = 0.25$, $\text{mass} = 6.7317$, $\text{relative error} = 0.005139$.

Mid-left, $T = 0.5$, $\text{mass} = 6.6841$, $\text{relative error} = 0.012178$.

Mid-right, $T = 0.75$, $\text{mass} = 6.6466$, $\text{relative error} = 0.017721$.

Right, $T = 1$, $\text{mass} = 6.6197$, $\text{relative error} = 0.021695$.

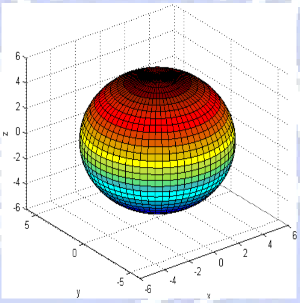


Figure 107: $T = 5$, $\text{mass} = 6.5261$, $\text{relative error} = 0.035528$.

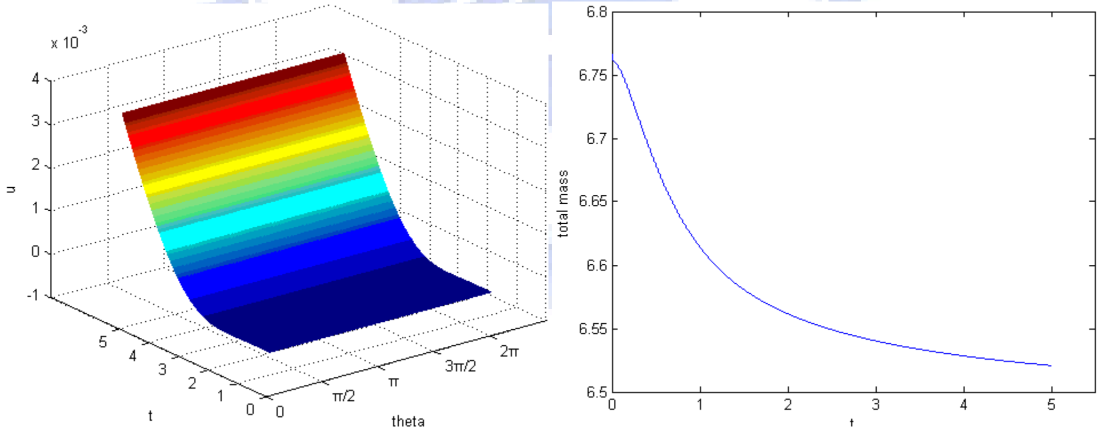


Figure 108: Left, the values of u on the equator change with time.

Right, total mass of u changes with time in $0 \sim 5$ seconds.

According to Figure 106, Figure 107, Figure 108, u was diffused as time goes by to be like the look of the original domain. Total mass is getting decreasing, but it will turn to stable that means mass will not change any more in a long time. Case 3 does not comply with the mass conservation law.

Case 4: Initial setting is the same as Figure 20,

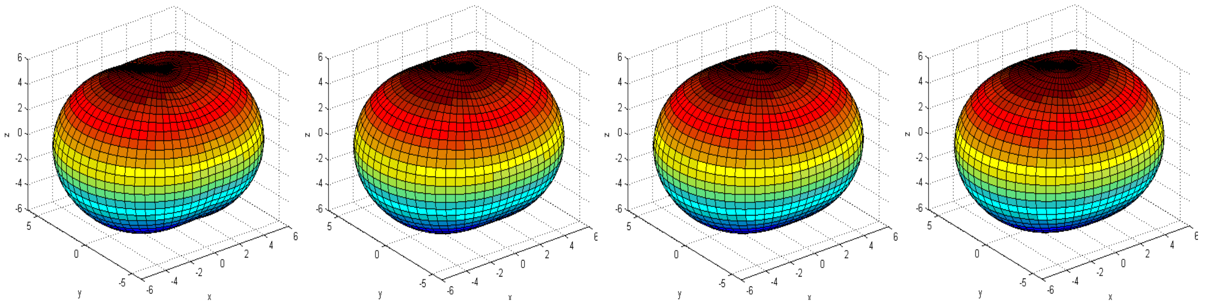


Figure 109: Left, T = 0.25, mass = 446.7227, relative error = 0.00014096.
Mid-left, T = 0.5, mass = 446.7604, relative error = 0.00022546.
Mid-right, T = 0.75, mass = 446.7885, relative error = 0.00028826.
Right, T = 1, mass = 446.8117, relative error = 0.00034017.

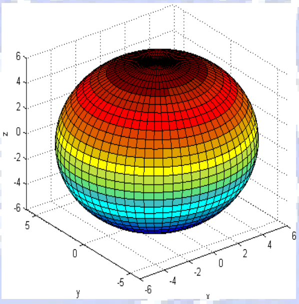


Figure 110: T = 5, mass = 446.9889, relative error = 0.00073693.

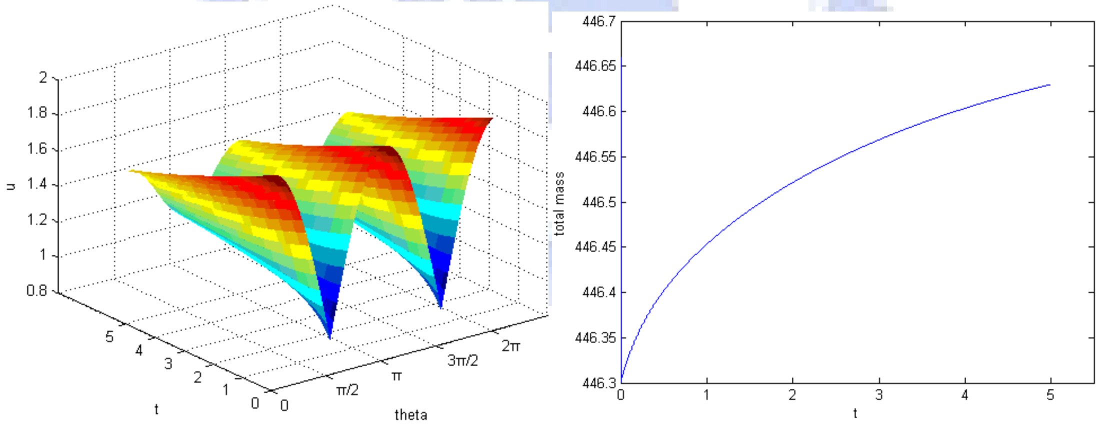


Figure 111: Left, the values of u on the equator change with time.
Right, total mass of u changes with time in 0~5 seconds.

According to Figure 109, Figure 110, Figure 111, u was diffused as time goes by to be like the look of the original domain. Total mass is getting increasing, but it will turn to stable that means mass will not change any more in a long time. Case 4 does not comply with the mass conservation law.

In above Case 1~Case 4, we derive that Case 1 can get the better result by mass conservation with some machine error；the others show up a little bit derivation that is less than one percent in our relative error.

6.4 Comparison

According to the results in Section 6.3.1 and Section 6.3.2, we could summarize the following table.

	6.3.1	6.3.2
Case1	1.8856e-005	1.8826e-005
Case2	0.0012786	0.001275
Case3	0.035539	0.035528
Case4	0.00073756	0.00073693

Table 11: The relative error in different cases with different methods at $\mathrm{T} = 5$.

Table 11 brings us that the solvers on the spherical surface domain which of Section 6.3.1 or the solvers which of Section 6.3.2 are coming to the same thing here. Thus, our solver in the next section will use Section 6.3.1 and it will save our time in writing the programming.

7、Numerical Solver of the Convection-Diffusion Equation

Here we solve the convection-diffusion equation on the moving domain by the discretization in Chapter 6 and observe how the total mass changes as time goes by in these solvers.

7.1 Discretization of the Convection-Diffusion Equation

The convection-diffusion equation can be described as

$$\frac{\partial \Gamma}{\partial \mathrm{t}} + (\nabla_{\mathrm{s}} \cdot \mathbf{u})\Gamma = \frac{1}{\mathrm{Pe_s}} \Delta_{\mathrm{s}} \Gamma \tag{47}$$

$\mathrm{Pe_s}$ means the Peclet number on the surface domain, and we set it be constant. Γ could be thought as the concentration or some things which have density. Moving surface domain that we given can be described as

$$\begin{cases} \dfrac{\partial \mathbf{X}(\mathrm{r},\phi,\theta,\mathrm{t})}{\partial \mathrm{t}} = \mathbf{u}(\mathrm{r},\phi,\theta,,\mathrm{t}) \\ \mathbf{X}(\mathrm{r},\phi,\theta,0) = (\mathrm{rsin}\phi\mathrm{cos}\theta,\mathrm{rsin}\phi\mathrm{sin}\theta,\mathrm{rcos}\phi) \end{cases}$$

means that the domain is moving with the velocity $\mathbf{u}$, and the original domain is spherical surface domain with $\mathrm{R} = 5$. Given $\mathbf{u}$ as follows,

$$\mathbf{u} = (\mathrm{zcos}(2\pi\mathrm{t}),0,0)$$

where $\mathbf{u}$ satisfies

$$\nabla \cdot \mathbf{u} = 0$$

[8] gives us

$$(\nabla_{\mathrm{s}} \cdot \mathbf{u}) = \frac{\dfrac{\partial}{\partial \mathrm{t}}\left(\left|\dfrac{\partial \mathbf{X}}{\partial \phi} \times \dfrac{\partial \mathbf{X}}{\partial \theta}\right|\right)}{\left|\dfrac{\partial \mathbf{X}}{\partial \phi} \times \dfrac{\partial \mathbf{X}}{\partial \theta}\right|}$$

Substitute into (47),

$$\frac{\partial \Gamma}{\partial \mathrm{t}} + \frac{\dfrac{\partial}{\partial \mathrm{t}}\left(\left|\dfrac{\partial \mathbf{X}}{\partial \phi} \times \dfrac{\partial \mathbf{X}}{\partial \theta}\right|\right)}{\left|\dfrac{\partial \mathbf{X}}{\partial \phi} \times \dfrac{\partial \mathbf{X}}{\partial \theta}\right|}\Gamma = \frac{1}{\mathrm{Pe_s}}\Delta_{\mathrm{s}}\Gamma$$

Product $\left|\frac{\partial \mathbf{X}}{\partial \phi} \times \frac{\partial \mathbf{X}}{\partial \theta}\right|$ to both sides on the above equation,

$$\frac{\partial \Gamma}{\partial \mathrm{t}}\left|\frac{\partial \mathbf{X}}{\partial \phi} \times \frac{\partial \mathbf{X}}{\partial \theta}\right| + \frac{\partial}{\partial \mathrm{t}}\left(\left|\frac{\partial \mathbf{X}}{\partial \phi} \times \frac{\partial \mathbf{X}}{\partial \theta}\right|\right)\Gamma = \frac{1}{\mathrm{Pe_s}}\Delta_{\mathrm{s}}\Gamma\left|\frac{\partial \mathbf{X}}{\partial \phi} \times \frac{\partial \mathbf{X}}{\partial \theta}\right| \qquad (48)$$

Use the Crank-Nicolson method to the left hand side of (48),

$$\frac{\partial \Gamma}{\partial \mathrm{t}}\left|\frac{\partial \mathbf{X}}{\partial \phi} \times \frac{\partial \mathbf{X}}{\partial \theta}\right| + \frac{\partial}{\partial \mathrm{t}}\left(\left|\frac{\partial \mathbf{X}}{\partial \phi} \times \frac{\partial \mathbf{X}}{\partial \theta}\right|\right)\Gamma$$

$$= \left(\frac{\Gamma^{\mathrm{n}+1} - \Gamma^{\mathrm{n}}}{\Delta \mathrm{t}}\right)\frac{\left|\dfrac{\partial \mathbf{X}^{\mathrm{n}+1}}{\partial \phi} \times \dfrac{\partial \mathbf{X}^{\mathrm{n}+1}}{\partial \theta}\right| + \left|\dfrac{\partial \mathbf{X}^{\mathrm{n}}}{\partial \phi} \times \dfrac{\partial \mathbf{X}^{\mathrm{n}}}{\partial \theta}\right|}{2}$$

$$+ \left(\frac{\left|\dfrac{\partial \mathbf{X}^{\mathrm{n}+1}}{\partial \phi} \times \dfrac{\partial \mathbf{X}^{\mathrm{n}+1}}{\partial \theta}\right| - \left|\dfrac{\partial \mathbf{X}^{\mathrm{n}}}{\partial \phi} \times \dfrac{\partial \mathbf{X}^{\mathrm{n}}}{\partial \theta}\right|}{\Delta \mathrm{t}}\right)\frac{\Gamma^{\mathrm{n}+1} + \Gamma^{\mathrm{n}}}{2}$$

$$= \frac{\Gamma^{\mathrm{n}+1}}{\Delta \mathrm{t}}\left|\frac{\partial \mathbf{X}^{\mathrm{n}+1}}{\partial \phi} \times \frac{\partial \mathbf{X}^{\mathrm{n}+1}}{\partial \theta}\right| - \frac{\Gamma^{\mathrm{n}}}{\Delta \mathrm{t}}\left|\frac{\partial \mathbf{X}^{\mathrm{n}}}{\partial \phi} \times \frac{\partial \mathbf{X}^{\mathrm{n}}}{\partial \theta}\right|$$

Use the explicit method to the right hand side of (48), and it becomes

$$\frac{1}{\mathrm{Pe_s}}\Delta_s\Gamma^n\left|\frac{\partial\mathbf{X}^n}{\partial\phi}\times\frac{\partial\mathbf{X}^n}{\partial\theta}\right|$$

Hence, in this section it equals to solve

$$\frac{\Gamma^{n+1}}{\Delta t}\left|\frac{\partial\mathbf{X}^{n+1}}{\partial\phi}\times\frac{\partial\mathbf{X}^{n+1}}{\partial\theta}\right|-\frac{\Gamma^n}{\Delta t}\left|\frac{\partial\mathbf{X}^n}{\partial\phi}\times\frac{\partial\mathbf{X}^n}{\partial\theta}\right|=\frac{1}{\mathrm{Pe_s}}\Delta_s\Gamma^n\left|\frac{\partial\mathbf{X}^n}{\partial\phi}\times\frac{\partial\mathbf{X}^n}{\partial\theta}\right|$$

i.e.

$$\Gamma^{n+1}=\frac{\left(\frac{1}{\mathrm{Pe_s}}\Delta_s\Gamma^n\left|\frac{\partial\mathbf{X}^n}{\partial\phi}\times\frac{\partial\mathbf{X}^n}{\partial\theta}\right|+\frac{\Gamma^n}{\Delta t}\left|\frac{\partial\mathbf{X}^n}{\partial\phi}\times\frac{\partial\mathbf{X}^n}{\partial\theta}\right|\right)}{\left|\frac{\partial\mathbf{X}^{n+1}}{\partial\phi}\times\frac{\partial\mathbf{X}^{n+1}}{\partial\theta}\right|}\Delta t \qquad (49)$$

Our grids, discretization and techniques are the same with Section 6.1 and central difference method 1 in Section 6.1.

$$\frac{\partial\mathbf{X}_{i,j}}{\partial\phi}\approx\frac{\mathbf{X}_{i+1,j}-\mathbf{X}_{i-1,j}}{2\Delta\phi}\,,\quad\frac{\partial\mathbf{X}_{i,j}}{\partial\theta}\approx\frac{\mathbf{X}_{i,j+1}-\mathbf{X}_{i,j-1}}{2\Delta\theta}$$

$$\frac{\partial\Gamma_{i,j}}{\partial\phi}\approx\frac{\Gamma_{i+1,j}-\Gamma_{i-1,j}}{2\Delta\phi}\,,\quad\frac{\partial\Gamma_{i,j}}{\partial\theta}\approx\frac{\Gamma_{i,j+1}-\Gamma_{i,j-1}}{2\Delta\theta}$$

7.2 Results of the Convection-Diffusion Equation on the Moving Surface

Because we have no exact solution of (48) in Section 7.1, we observe how the concentration goes in our numerical solver as time goes by to check if it makes sense. And consider that the total mass change with time passing here. Set $\Gamma\geq 0$ everywhere. We computed the total mass as Section 6.3

$$\sum_{\substack{i=1,\dots,M\\ j=1,\dots,N}} u_{i,j}\left|\frac{\partial\mathbf{X}_{i,j}}{\partial\phi}\times\frac{\partial\mathbf{X}_{i,j}}{\partial\theta}\right|\Delta\phi\Delta\theta$$

and our initial total mass is like Section 3.3,

$$S^0=\sum_{\substack{i=1,\dots,M\\ j=1,\dots,N}} u_{i,j}^0R^2|\sin\phi_i|\,\Delta\phi\Delta\theta$$

Use the solver of (45) in (47), $\Gamma_t=\Delta_s\Gamma$ on the spherical surface domain by Section 6.1 with central difference method 1. Then, we could solve (49) easily and the results are as follows. Initial settings：$M=32, N=64,\ R=5,\ \Delta t=\frac{1}{2}\frac{1}{(2\pi)^2}\Delta x^2=\frac{1}{2}\frac{1}{N^2}$. Initial domain is the same as Figure 7.

Case 1: Initial setting is the same as Figure 8,

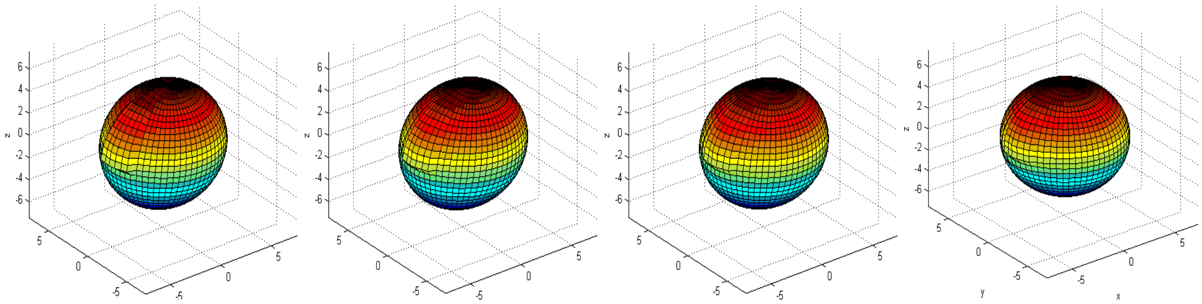


Figure 112: Left, T = 0.125, mass = 14.6845, relative error = 0.0032324.
Mid-left, T = 0.25, mass = 14.6845, relative error = 0.0032323.
Mid-right, T = 0.375, mass = 14.6846, relative error = 0.0032264.
Right, T = 0.5, mass = 14.6846, relative error = 0.0032238.

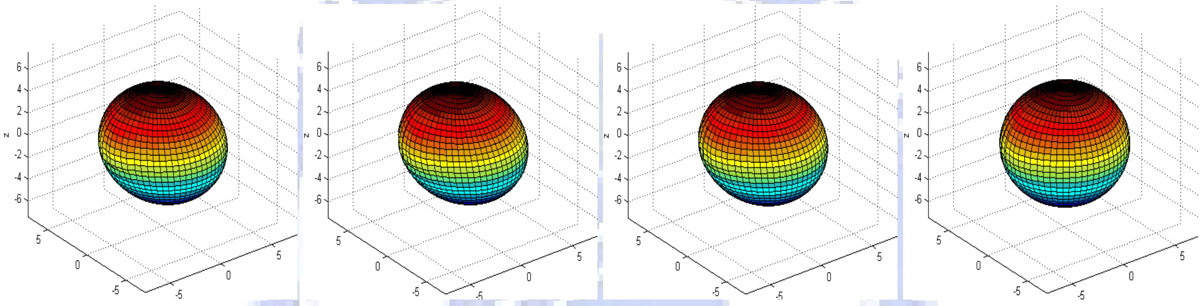


Figure 113: Left, T = 0.625, mass = 14.6847, relative error = 0.00322.
Mid-left, T = 0.75, mass = 14.6849, relative error = 0.0032077.
Mid-right, T = 0.875, mass = 14.685, relative error = 0.003197.
Right, T = 1, mass = 14.685, relative error = 0.0031976.

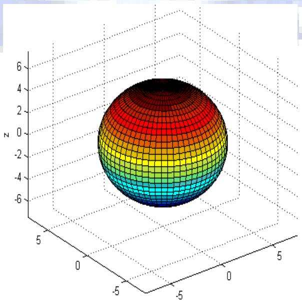


Figure 114: T = 5, mass = 14.6851, relative error = 0.0031911.

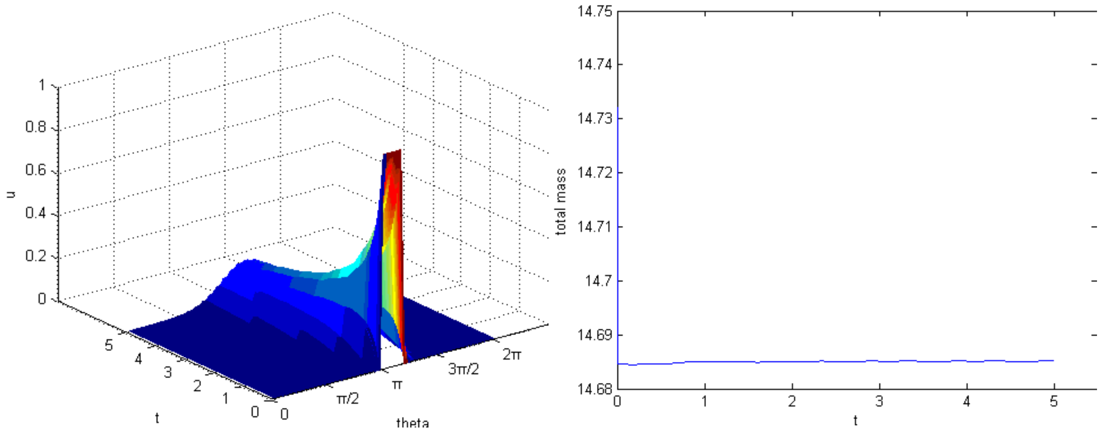


Figure 115： Left, the values of u on the equator change with time.

Right, total mass of u changes with time in $0 \sim 5$ seconds.

According to Figure 112, Figure 113, Figure 114, Figure 115, u was diffused as time goes by to be like the look of the original domain. Total mass is getting increasing a little bit, but it will turn to stable that means mass will not change any more in a long time. Case 1 does not comply with the mass conservation law.

Case 2: Initial setting is the same as Figure 12,

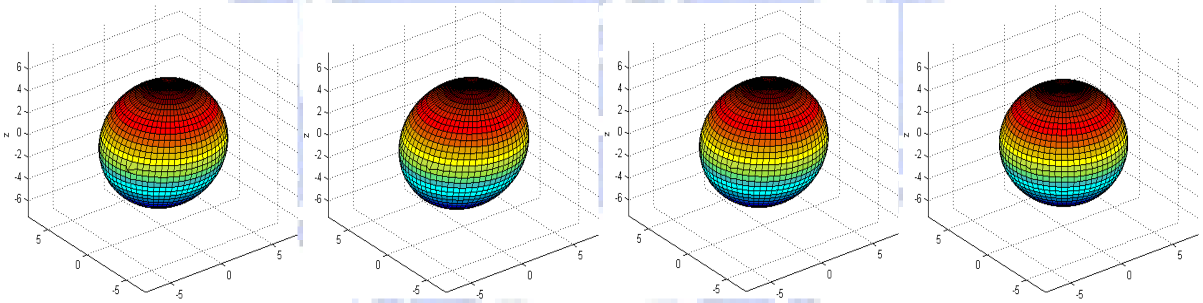


Figure 116： Left, $T = 0.125$, $mass = 0.2399$, $relative\ error = 0.0031928$.

Mid-left, $T = 0.25$, $mass = 0.23991$, $relative\ error = 0.0031652$.

Mid-right, $T = 0.375$, $mass = 0.23991$, $relative\ error = 0.0031418$.

Right, $T = 0.5$, $mass = 0.23991$, $relative\ error = 0.0031313$.

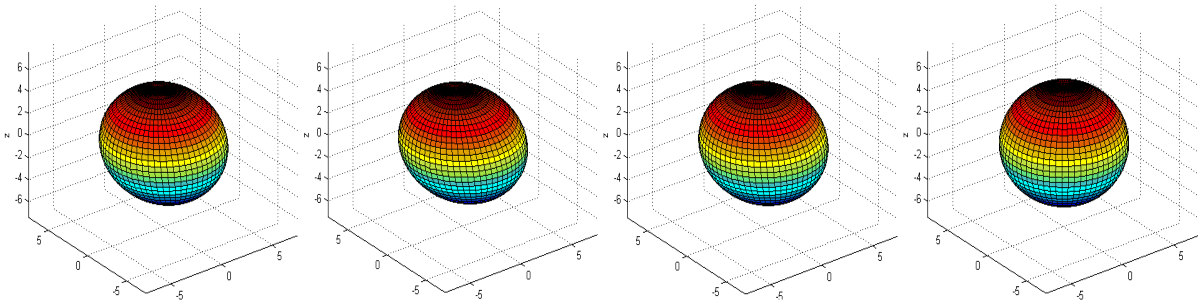


Figure 117： Left, $T = 0.625$, $mass = 0.23992$, $relative\ error = 0.0031201$.

Mid-left, $T = 0.75$, $mass = 0.23992$, $relative\ error = 0.0030963$.

Mid-right, T = 0.875, mass = 0.23993, relative error = 0.0030691.
Right, T = 1, mass = 0.23993, relative error = 0.0030483.

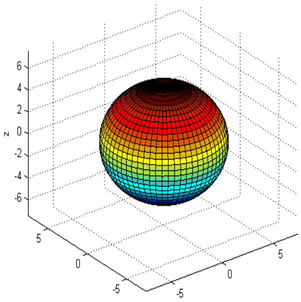


Figure 118: T = 5, mass = 0.24019, relative error = 0.0019839.

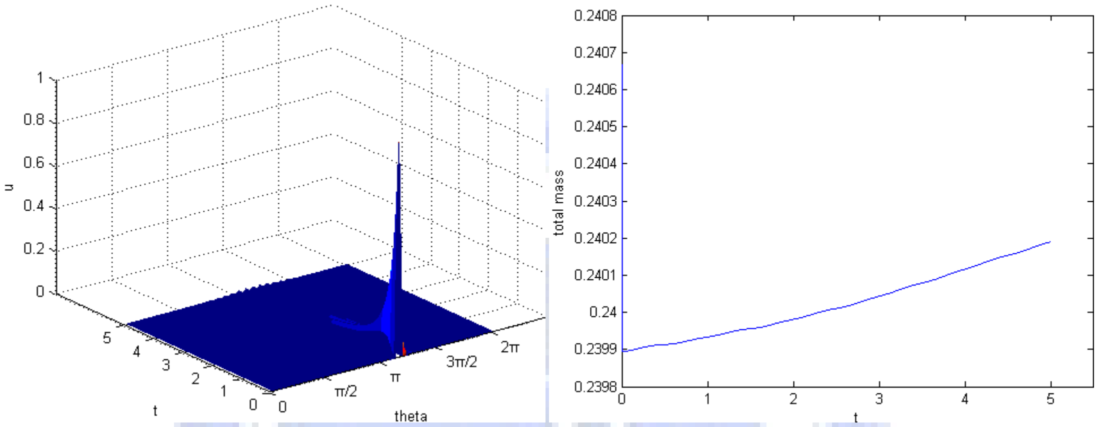


Figure 119: Left, the values of u on the equator change with time.
Right, total mass of u changes with time in 0~5 seconds.

According to Figure 116, Figure 117, Figure 118, Figure 119, u was diffused as time goes by to be like the look of the original domain. Total mass is getting increasing a little bit, but it will turn to stable that means mass will not change any more in a long time. Case 2 does not comply with the mass conservation law.

Case 3: Initial setting is the same as Figure 16,

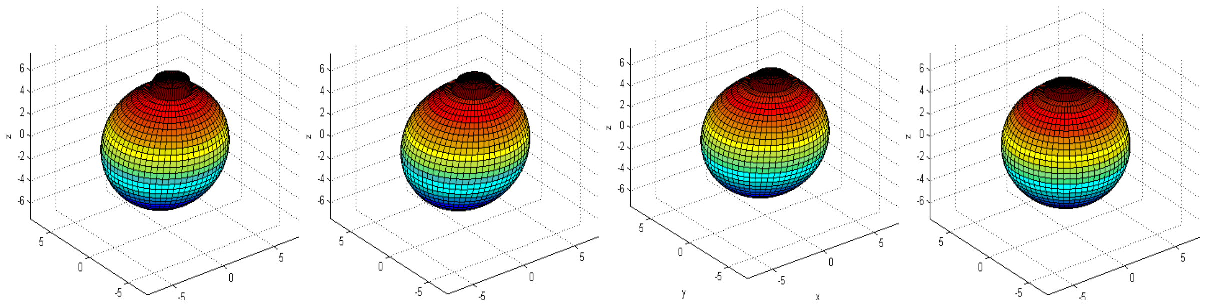


Figure 120: Left, T = 0.125, mass = 6.7328, relative error = 0.0049763.
Mid-left, T = 0.25, mass = 6.71, relative error = 0.008356.
Mid-right, T = 0.375, mass = 6.6853, relative error = 0.012002.

Right, T = 0.5, mass = 6.6623, relative error = 0.015401.

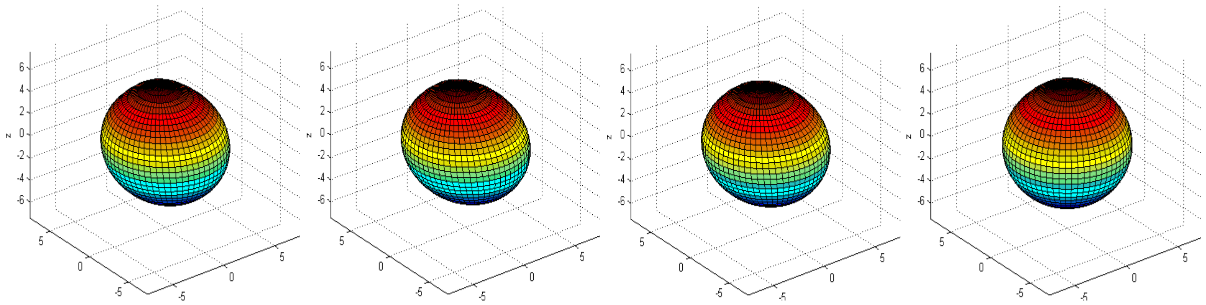


Figure 121: Left, T = 0.625, mass = 6.6421, relative error = 0.018383.
Mid-left, T = 0.75, mass = 6.6249, relative error = 0.020933.
Mid-right, T = 0.875, mass = 6.6103, relative error = 0.02309.
Right, T = 1, mass = 6.598, relative error = 0.024904.

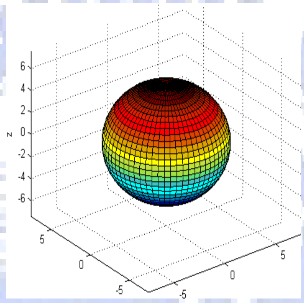


Figure 122: Left, T = 5, mass = 6.5046 relative error = 0.0387.

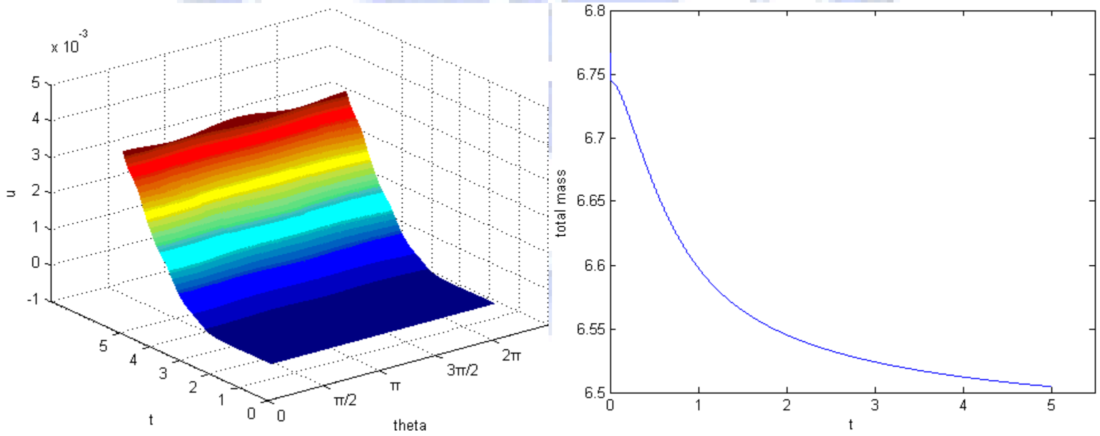


Figure 123: Left, the values of u on the equator change with time.
Right, total mass of u changes with time in 0~5 seconds.

According to Figure 120, Figure 121, Figure 122, Figure 123, u was diffused as time goes by to be like the look of the original domain. Total mass is getting decreasing, but it will turn to stable that means mass will not change any more in a long time. Case 3 does not comply with the mass conservation law.

Case 4: Initial setting is the same as Figure 20,

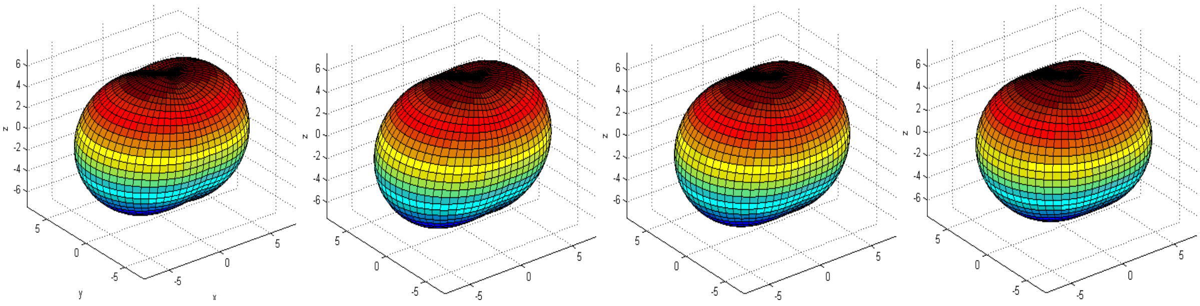


Figure 124: Left, T = 0.125, mass = 445.262, relative error = 0.0031292.
Mid-left, T = 0.25, mass = 445.2899, relative error = 0.0030667.
Mid-right, T = 0.375, mass = 445.3121, relative error = 0.0030172.
Right, T = 0.5, mass = 445.3128, relative error = 0.0029813.

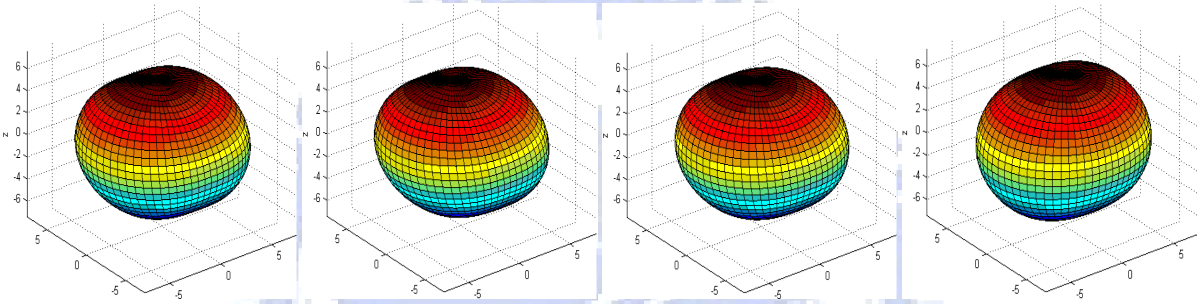


Figure 125: Left, T = 0.625, mass = 445.3418, relative error = 0.0029505.
Mid-left, T = 0.75, mass = 445.3565, relative error = 0.0029178.
Mid-right, T = 0.875, mass = 445.3699, relative error = 0.0028876.
Right, T = 1, mass = 445.38, relative error = 0.0028651.

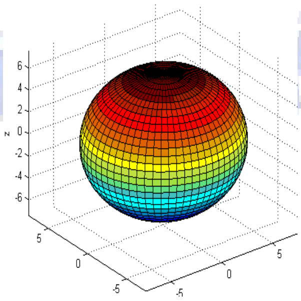


Figure 126: T = 5, mass = 446.56, relative error = 0.002462.

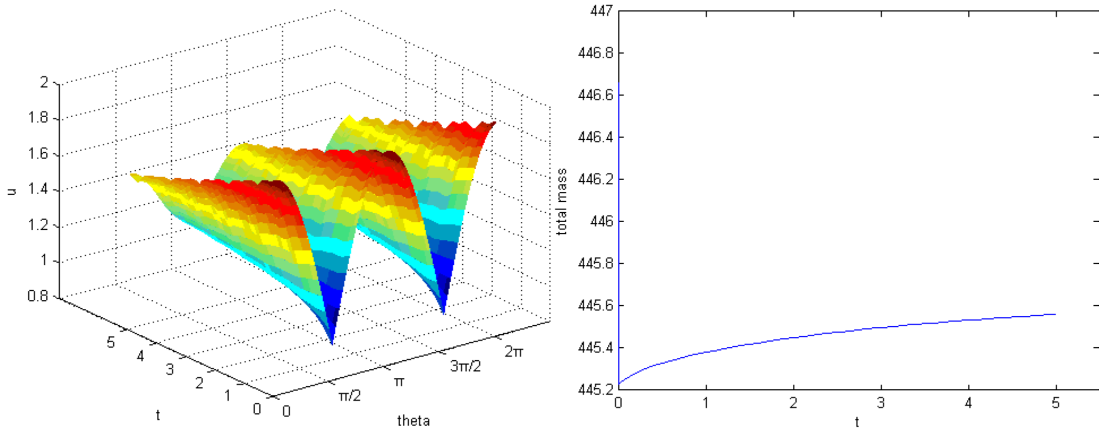


Figure 127： Left, the values of u on the equator change with time.
Right, total mass of u changes with time in $0 \sim 5$ seconds.

According to Figure 124, Figure 125, Figure 126, Figure 127, u was diffused as time goes by to be like the look of the original domain. Total mass is getting increasing, but it will turn to stable that means mass will not change any more in a long time. Case 4 does not comply with the mass conservation law. In above Case 1~Case 4, all cases show up a little bit derivation that is less than three percent in our relative error.

8、Applications

In our thesis, we have constructed many kinds of numerical solvers. We could apply them in physics, engineering and aerobiology, etc. The earth is considered a spherical surface. If we ignore the geography and wind direction, then we can predict the diffusion of noxious gas or pollutants, radiations, etc, on the earth. And the dissolution of glaciers affects the nearby environment on the earth is also an important application in our solver in Section 3.3.2 which complied with the mass conservation law. The applications in Chapter 4 are the same as Chapter 3 if the earth is considered an ellipsoid surface. The difference between Chapter 3 and Chapter 4 is only in the mass. We apply the solvers in Chapter 3 and Chapter 4 in Chapter 5. They could help solving the parabolic type equation which is like $u_t = \Delta u + \blacksquare$ on the spherical or ellipsoid surface domain. Apply Chapter 5 in the electrochemical waves of cardiac simulation in human heart. The solver in Chapter 6 could help solving the parabolic type equation which is like $\Gamma_t = \Delta\Gamma + \blacksquare$ on the moving surface if we modify the initial condition on the concentration Γ and domain $\mathbf{X}$. We apply the solvers in Chapter 7. Furthermore, we could solve the parabolic type equations on the moving domain with the initial domain which is ellipsoid, torus surface domain or something else. It will be very useful in the future in many applications.

9、Conclusion and Future Works

After discussing our paper in detail, Chapter 3 shows us that symmetric discretizaion is the better numerical method in the heat equation on the spherical surface domain in spherical coordinates about the mass conservation. Chapter 4 performs a fast heat solver on the ellipsoid surface domain in ellipsoid coordinates and we use it to help solving the reaction-diffusion equation, i.e. simulating the motions of the electrochemical waves in our heart with faster computations in Chapter 5. But the results in Chapter 5 are still far away from the realistic human hearts because the domain in human hearts is moving all the time. In Chapter 6 and Chapter 7, discuss the numerical solvers of the equations on the moving domain. Total mass of them will not change too much. In our future works, we hope to develop a scheme which combines forthcoming cardiac simulation as Chapter 5 and couples them with moving domain. There is still a big problem need to settle is constructing a moving equation which satisfies the realistic heart. After setting them all, we could impose our methods as Chapter 7 which solve the equations on the moving domain on simulation of hearts and get more authentic simulations. In summary, the results we have done in Chapter 5 are the propagation of electrochemical waves in human heart is an especially important example. But it is known that factors such as topology, thickness, and differences of cardiac tissue can strongly influence wave propagation [1]. This is another difficulty that must be conquered.

Reference

[1] Amdjadi F., Gomatam J., Spiral waves on static and moving spherical domain, Journal

of Computional and Applied Mathematics. Vol. 182 , 472~486, (2005).

[2] Barkley, D., A model for fast computer simulation of waves in excitable media, Phys D49, 61-70, 1991.

[3] Barkley, D., Kness, M. and Tuckerman, L. S., Spiral-wave dynamics in a simple model of excitable media：The transition from simple to compound rotation. Physical Review A , Vol.42, Nos.4, 2489-2492, 1990.

[4] Cooley,J.W. and Tukey,J.W.,“An Algorithm for the Machine Calculation of Complex Fourier Series.”Math. Comp. 19, 297-301 (1965).

[5] Danaila,I., Joly, P., Kaber, S.M., Postel, M., An Introduction to Scientific Computing. Springer.

[6] Gomatam, J.and Amdjadi, F., Reaction-diffusion equations on a sphere：Meandering of spiral waves. Physical Review E, Vol.56, Nos.4, 3913-3919, 1997.

[7] Huang, H. and Lai, M.C., A note on decoupled jump conditions for Navier-Stokes flows with discontinuous viscosity across an incompressible membrane.

[8] Huang, H., Lai, M.C. and Tseng Y.H., An immersed boundary method for interfacial flows with insoluble surfactant.

[9] Isaacson, E. and Keller, H.B., Analysis of numerical methods. 1966.

[10] Lai, M.C., A fast spectral/difference method without pole conditions for Poisson-type equations in cylindrical and spherical geometries. IMA J. of Numerical Analysis, vol 22, No 4, 537-548, (2002).

[11] Lai, M.C., Fast Poisson Solver in a Three-dimensional Ellipsoid. Contemporary Mathematics, AMS, Vol.329, 203-208, 2003.

[12] Lai, M.C. and Wang, W.C.,Fast direct solvers for Poisson equation on 2D polar and spherical geometries. Numerical Methods for Partial Differential Equations, vol 18, 56-68, (2002).

[13] Ralph Lewis, H., Bellan, P.M., Physical constraints on the coefficients of Fourier

expansions in cylindrical coordinates.J Math Phys. Vol.31, Nos.11, 2592-2596, 1990.

[14] SAUER,T., Numerical analysis.Pearson Education.

[15] Winfree A.T., When Time Breaks Down, Princeton University Press,Princeton, NJ, 1987.

[16] Xu, J.J. and Zhao H.K., An Eulerian Formulation for Solving Partial Differential Equations along a Moving Interface. Journal of Scientific Computing, Vol.19, Nos.1-3, 573-594, 2003.