

# 螺旋波的产生及混沌控制

俞炆

2020.6.16

## 摘要

文章以 FitzHugh-Nagumo 模型为基础, 详细阐述了螺旋波及其湍流态的产生方法, 并利用行波法对其进行了有效控制。

关键词: 斑图动力学 螺旋波 行波法

## 1 螺旋波模型简介

### 1.1 数学表述

螺旋波通常产生于反应扩散系统。在本文中, 我们用一个描述在可兴奋性媒介中波动传播的模型来产生, 并控制螺旋波及其湍流态。此模型为 FitzHugh-Nagumo 模型, 其数学表述如下:

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

$$\frac{\partial v}{\partial t} = f(u) - v \quad (2)$$

$$(3)$$

其中:

$$f(u) = \begin{cases} 0 & , 0 \leq u < \frac{1}{3} \\ 1 - 6.75u(u-1)^2 & , \frac{1}{3} \leq u \leq 1 \\ 1 & , u > 1 \end{cases}$$

此模型给出的边界条件为零流边界条件, 即在边界上满足:

$$\frac{\partial u}{\partial n} = 0 \quad (4)$$

方程 1 中  $v$  的初值设置为 0;  $u$  在左边界上的初值为 1, 其余均为 0. 其余参数罗列如下:  $a=0.84, b=0.07, \varepsilon \in [0, 0.07]$  (非混沌状态)。

### 1.2 模型的 Matlab 求解与演示方法

本文中将对三个问题进行处理和讨论:

- 行波与靶波的产生
- 螺旋波的产生

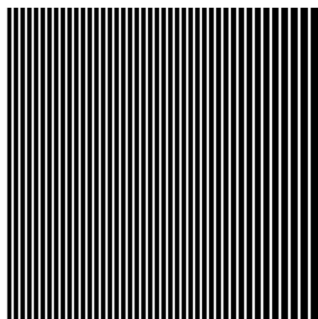
- 混沌现象的产生
- 使用行波法实现螺旋波与混沌现象的控制

以上四个问题采用的方法为显式欧拉法，其中时间步长为 0.05，空间格点间距选为 0.5；

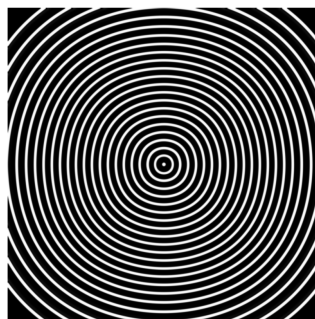
对于前三个问题，本文在  $800 \times 800$  的格点上对该模型的不同条件进行求解并实现可视化效果。由于最后一步在尝试过程中发现  $800 \times 800$  格点上行波对于混沌/螺旋波的控制过程所需时间过长，因此采用  $100 \times 100$  格点进行实现。

## 2 行波与靶波的产生

当我们对于方程的左边界施加一个持续性的刺激，即令左边界初值周期性输出 1 时，我们可得到行波的图像，如图 1(a)所示；如需产生靶波图像，只需在所有  $u$  的初值中间四个点设置为 1，其余  $u$  值为 0 即可。其具体图像如下图 1(b)所示：



(a) 行波



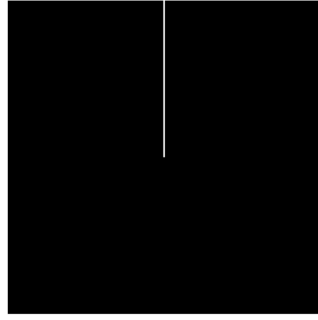
(b) 靶波

图 1: 在不同初值条件下产生的行波与靶波

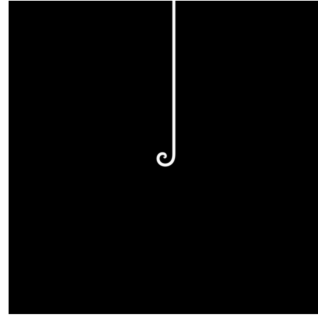
## 3 螺旋波的产生

如果在行波产生的过程中，将其突然“抹去”一半，则会导致行波传播的不均匀，从而导致了螺旋波的产生。

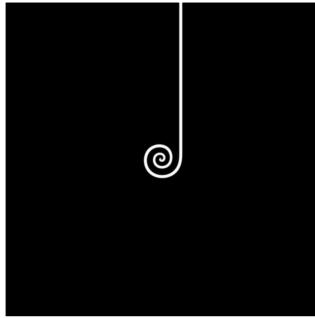
下图为螺旋波的产生过程，其中  $\varepsilon = 0.04$ ：图 2为螺旋波从 0.05ms 至 720ms 时的演化过程的六副截图。



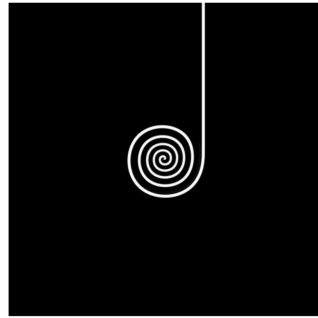
(a) 0.05ms



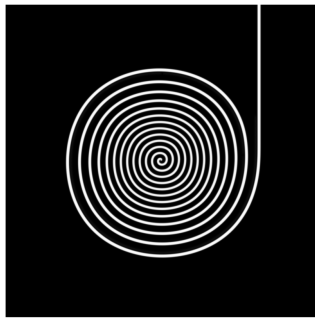
(b) 5ms



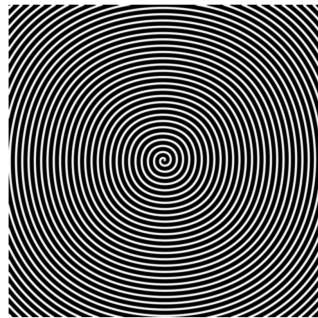
(c) 10ms



(d) 20ms



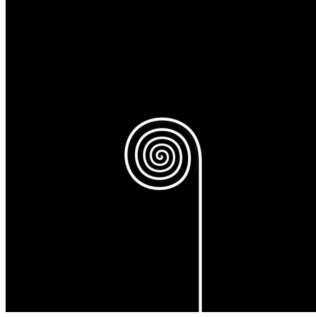
(e) 50ms



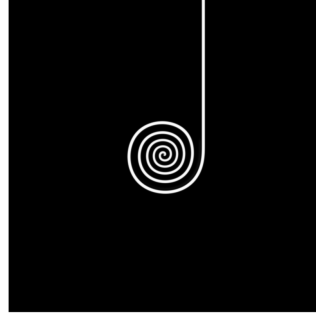
(f) 720ms

图 2: 不同时间下的螺旋波的形态演化

通过调整抹去上半部分或下半部分，我们可得到回旋方向不同的螺旋波，图 3(a)为抹去下半部分演化至 20ms 的螺旋波；图 3(b)为抹去上半部分演化至 20ms 的螺旋波。



(a) 抹去下半部分演化至 20ms 的螺旋波



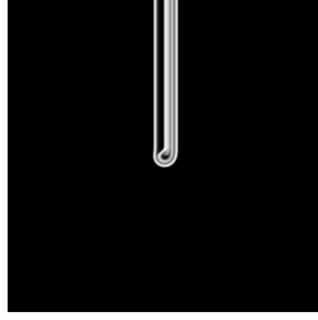
(b) 抹去上半部分演化至 20ms 的螺旋波

图 3: 抹去不同位置产生回旋方向不同的螺旋波

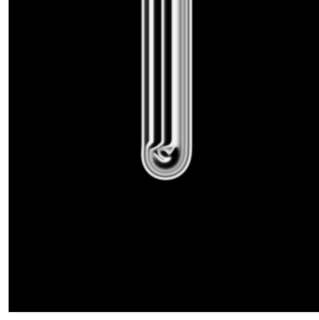
## 4 混沌状态

当  $\varepsilon \in [0, 0.07]$  时，我们能得到稳定状态下的螺旋波；而当  $\varepsilon > 0.07$  时，我们可明显观测到螺旋波失稳而产生湍流的现象。

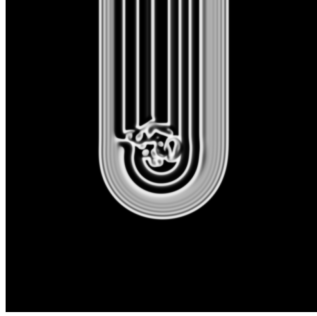
我们令  $\varepsilon = 0.1$ , 如图??所示，我们得到了在不同时间下产生时空混沌的图像。



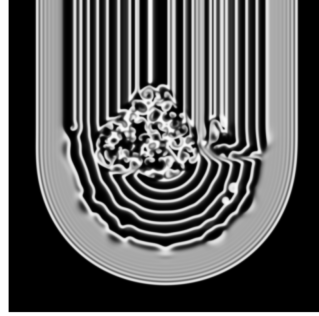
(a) 10ms



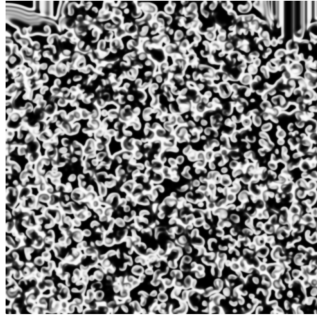
(b) 20ms



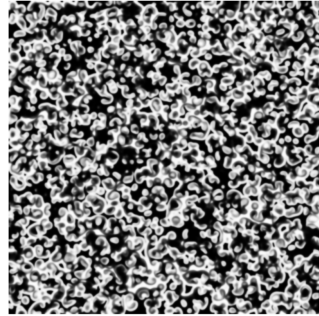
(c) 50ms



(d) 100ms



(e) 250ms



(f) 400ms

图 4: 不同时间下时空混沌的演化图像

## 5 使用行波法实现螺旋波与混沌现象的控制

### 5.1 控制原理

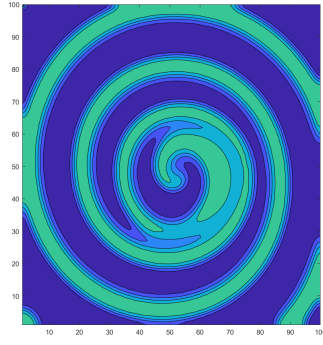
使用行波法实现螺旋波与混沌现象的控制的原理非常简单，即在螺旋波/混沌产生之后，不断利用行波冲击螺旋波，最中达到消除螺旋波/混沌，回到行波态的过程。以下为具体的过程演示，其中  $\varepsilon = 0.55$ ：

## 5.2 螺旋波的控制

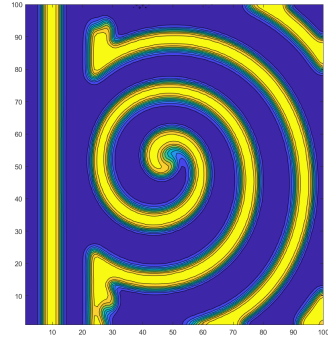
首先，我们按照上文螺旋波方法先产生一个螺旋波，在产生螺旋波之后。给左边的边界施加一个持续的刺激，即，使  $u$  周期性输出 0, 1，从而产生源源不断的行波。然后观察行波冲击螺旋波的结果。

下图 5 为在不同时间下螺旋波的产生与控制过程图像。从图中可知，控制过程比螺旋波产生过程长很多：形成一个螺旋波仅需 20ms, 而完全消除一个螺旋波需要 392.50ms。

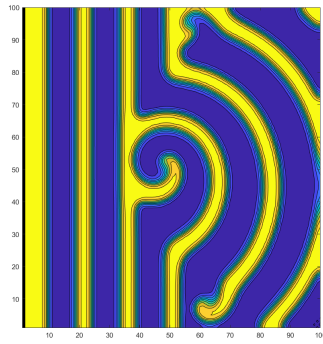
另外， $\varepsilon$  的值不同，也会导致螺旋波控制时间的不同。当  $\varepsilon = 0.04$  时，我们发现时间演化超过 1000ms 时行波依然没有一点控制螺旋波的迹象。此处有待进一步的研究探索。



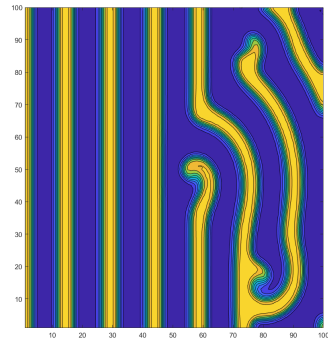
(a) 20ms 产生螺旋波



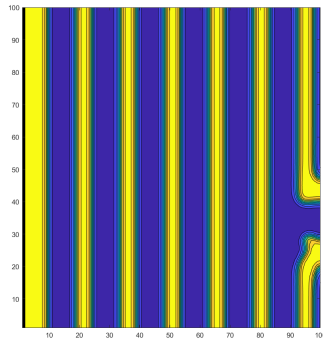
(b) 螺旋波产生后行波进行控制 105ms



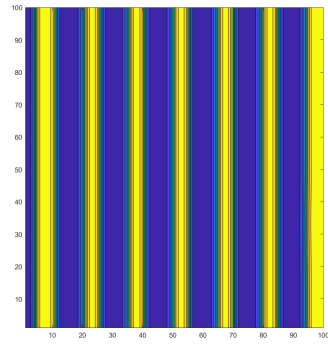
(c) 螺旋波产生后行波进行控制 200ms



(d) 螺旋波产生后行波进行控制 250ms



(e) 螺旋波产生后行波进行控制 372ms

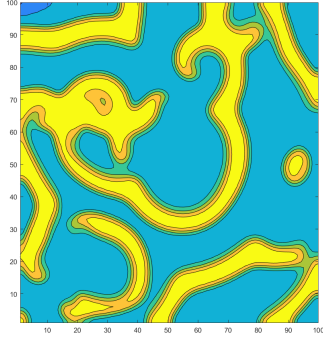


(f) 螺旋波产生后行波进行控制 392.50ms

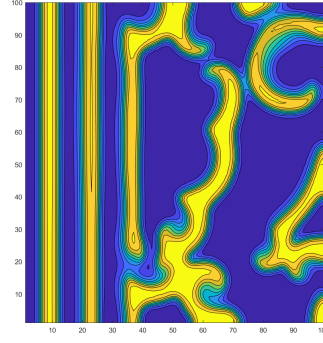
图 5: 螺旋波的控制

### 5.3 时空混沌的控制

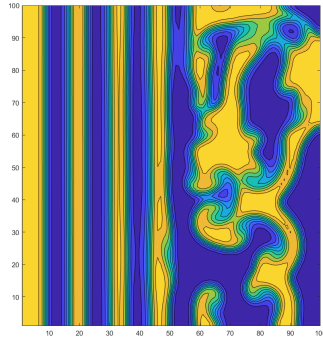
时空控制的方法与螺旋波控制的方法相同，两者像区别之处仅仅在于  $\varepsilon$  不同：我们此处实现时空混沌的控制时给出的  $\varepsilon = 0.0701$ 。下图 6 是时空混沌的产生以及控制过程中产生的图像：



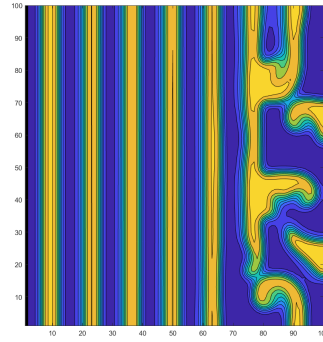
(a) 50ms 产生时空混沌



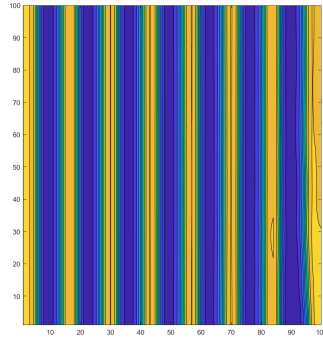
(b) 时空混沌产生后行波进行控制 65ms



(c) 时空混沌产生后行波进行控制 100ms



(d) 时空混沌产生后行波进行控制 165ms



(e) 时空混沌产生后行波进行控制 215ms

图 6: 时空混沌的控制

由图像演化过程可知：在  $\varepsilon = 0.701$  下产生的时空混沌被行波完全控制所需的时间比  $\varepsilon = 0.55$  的螺旋波被行波完全控制所需的时间要短；而且其行波控制时空混沌所需的时间仅仅需要 215ms，约为 2s，时间非常短，可知用行波对心脏病入进行控制治疗，从而使时空混沌消失前景十分广阔。



## 6 总结

本文基于 FitzHugh-Nagumo 模型, 对螺旋波及其湍流态的产生以及控制做了相关的计算机模拟。在模拟过程中, 我们发现使用行波对于时空混沌进行控制所需时间极短, 且其操作并不复杂, 便于实现。如将其运用于心脏病人的心率恢复或者其他医学领域, 可能会有意想不到的效果, 具有非常广阔的应用前景。

## A 本次课题所需用到的所有的函数程序

%扩散方程的非齐次项

```
function y = f1(u,v)
global a b ep
y = -1/ep.*u.*(u-1).*(u-(v+b)./a);
end
```

% $f(u)$  函数

```
function y=f2(u)
global n1 n2
for i=1:n1
for j=1:n2
if u(i,j)>=0 && u(i,j)<1/3
y(i,j)=0;
elseif u(i,j)>=1/3 && u(i,j)<=1
y(i,j)=1-6.75*u(i,j)*(u(i,j)-1)^2;
elseif u(i,j)>1
y(i,j)=1;
end
end
end
```

%拉普拉斯算子对  $u$  作用

```
function la=laplace(u)
global n1 n2
%Euler法求  $u$ %
for i=2:n1-1
for j=2:n2-1
la(i,j)=u(i+1,j)+u(i-1,j)+u(i,j+1)+u(i,j-1)-4*u(i,j);
end
end
% $u$  值的初始条件%
la(1,1)=2.*u(1,2)+2.*u(2,1)-4.*u(1,1);
```

```

la(1,n2)=2*u(1,n2-1)+2*u(2,n2)-4*u(1,n2);
la(n1,1)=2*u(n1,2)+2*u(n1-1,1)-4*u(n1,n2);
la(n1,n2)=2*u(n1,n2-1)+2*u(n1-1,n2)-4*u(n1,n2);

for j=2:n2-1
    la(1,j)=u(1,j-1)+u(1,j+1)+2*u(2,j)-4*u(1,j);
    la(n1,j)=u(n1,j-1)+u(n1,j+1)+2*u(n1-1,j)-4*u(n1,j);
end

for i=2:n1-1
    la(i,1)=u(i-1,1)+u(i+1,1)+2*u(i,2)-4*u(i,1);
    la(i,n2)=u(i-1,n2)+u(i+1,n2)+2*u(i,n2-1)-4*u(i,n2);
end
end

```

## B 行波代码

```

clear all;clc;
%参数数值%
global n1 n2 la a b ep
a=0.84;
b=0.07;
n1=800;%x格点数
n2=800;%y格点数
ep=0.04;
la=zeros(n1,n2);
%拉普拉斯算符对u作用后的离散值
u=zeros(n1,n2);
%u函数离散值(初始)
v=zeros(n1,n2);
%v函数离散值(初始)
t=0;%时间初始值
dt=0.05;%时间间隔
N=60000;%时间点数
h=0.5;%相邻格点间隔

%循环遍历求得u的数值解并画图%
for i=1:N%时间循环步进
    t=t+dt;
    u=u+dt.*(f1(u,v)+laplace(u)/h^2);

```

```

v=v+dt.*(f2(u)-v);
if mod(t,4)>1
u(:,1)=1;
else
u(:,1)=0;
end
imagesc(u);
imshow(u);
end

```

## C 靶波代码

```

clear all;clc;
%参数数值%
global n1 n2 la a b ep
a=0.84;
b=0.07;
n1=800;%x格点数
n2=800;%y格点数
ep=0.04;
la=zeros(n1,n2);
%拉普拉斯算符对u作用后的离散值
u=zeros(n1,n2);
%u函数离散值(初始)
v=zeros(n1,n2);
%v函数离散值(初始)
t=0;%时间初始值
dt=0.05;%时间间隔
N=60000;%时间点数
h=0.5;%相邻格点间隔

%循环遍历求得u的数值解并画图%

for i=1:N%时间循环步进
t=t+dt;
u=u+dt.*(f1(u,v)+laplace(u)/h^2);
v=v+dt.*(f2(u)-v);
if mod(t,3)<1
u(399,399)=1;

```

```

u(399,401) = 1;
u(401,399) = 1;
u(401,401) = 1;
v(399,399) = 1;
v(399,401) = 1;
v(401,399) = 1;
v(401,401) = 1;
else
u(399,399) = 0;
u(399,401) = 0;
u(401,399) = 0;
u(401,401) = 0;
v(399,399) = 0;
v(399,401) = 0;
v(401,399) = 0;
v(401,401) = 0;
end
if t==100
m = 1;
end
%imshow(u);%画图
imshow(u);
xlabel('x'); ylabel('y');
end

```

## D 产生螺旋波的代码

```

clear all;clc;

%参数数值%

global n1 n2 la a b ep
a=0.84;
b=0.07;
n1=800;%x格点数
n2=800;%y格点数
ep=0.04;
la=zeros(n1,n2);
%拉普拉斯算符对u作用后的离散值
u=zeros(n1,n2);

```

```

%u函数离散值（初始）
v=zeros(n1,n2);
%v函数离散值（初始）
t=0;%时间初始值
dt=0.05;%时间间隔
N=60000;%时间点数
h=0.5;%相邻格点间隔
%螺旋波初条件%
u(401:800,398:402)=1;
v(401:800,394:398)=1;
for i=1:N
t=t+dt
u=u+dt.*(f1(u,v)+laplace(u)/h^2);
v=v+dt.*(f2(u)-v);
imshow(u);
end

```

## E 时空混沌的产生代码

```

clear all;clc;
%参数数值%
global n1 n2 la a b ep
a=0.84;
b=0.07;
n1=800;%x格点数
n2=800;%y格点数
ep=0.10;
la=zeros(n1,n2);
%拉普拉斯算符对u作用后的离散值
u=zeros(n1,n2);
%u函数离散值
v=zeros(n1,n2);
%v函数离散值
t=0;%时间初始值
dt=0.05;%时间间隔
N=60000;%时间点数
h=0.5;%相邻格点间隔

%螺旋波初条件%
u(1:400,398:402)=1;

```

```

v(1:400,394:398)=1;

%循环遍历求得u的数值解并画图%
for i=1:N%时间循环步进
t=t+dt
u=u+dt.*(f1(u,v)+laplace(u)/h^2);
v=v+dt.*(f2(u)-v);
imshow(u);
end

```

## F 控制螺旋波

```

clear all;clc;
%参数数值%
global n1 n2 la a b ep
a=0.84;
b=0.07;
n1=100;%x格点数
n2=100;%y格点数
ep=0.055;
la=zeros(n1,n2);
%拉普拉斯算符对u作用后的离散值
u=zeros(n1,n2);
%u函数离散值
v=zeros(n1,n2);
%v函数离散值
t=0;%时间初始值
dt=0.05;%时间间隔
N=60000;%时间点数
h=0.5;%相邻格点间隔

%螺旋波初条件%
u(1:50,48:52)=1;
v(1:50,44:48)=1;

%循环遍历求得u的数值解并画图%
for i=1:400%时间循环步进
t=t+dt
u=u+dt.*(f1(u,v)+laplace(u)/h^2);
v=v+dt.*(f2(u)-v);

```

```

%imshow(u)
contourf(u),pause(0.05);
end
t = 0;

for i=1:N%时间循环步进
t=t+dt
u=u+dt.*(f1(u,v)+laplace(u)/h^2);
v=v+dt.*(f2(u)-v);
if mod(t,4)>1
u(:,1)=1;
else
u(:,1)=0;
end
contourf(u),pause(0.05);
end

```

## G 段落

```

clear all;clc;
%参数数值%
global n1 n2 la a b ep
a=0.84;
b=0.07;
n1=100;%x格点数
n2=100;%y格点数
ep=0.0701;
la=zeros(n1,n2);
%拉普拉斯算符对u作用后的离散值
u=zeros(n1,n2);
%u函数离散值
v=zeros(n1,n2);
%v函数离散值
t=0;%时间初始值
dt=0.05;%时间间隔
N=60000;%时间点数
h=0.5;%相邻格点间隔

%%螺旋波初条件%
u(1:50,48:52)=1;

```

```

v(1:50,44:48)=1;

%循环遍历求得u的数值解并画图%
for i=1:1000%时间循环步进
t=t+dt
u=u+dt.*(f1(u,v)+laplace(u)/h^2);
v=v+dt.*(f2(u)-v);
%imshow(u)
contourf(u),pause(0.05);
end
t = 0;

for i=1:N%时间循环步进
t=t+dt
u=u+dt.*(f1(u,v)+laplace(u)/h^2);
v=v+dt.*(f2(u)-v);
if mod(t,4)>1
u(:,1)=1;
else
u(:,1)=0;
end
%plot(u);
%imshow(u);
contourf(u),pause(0.05);
end

```

## 参考文献

[1] 吴信谊 Morris-Lecar 神经元放电行为及螺旋波形成机制的研究 [D]. 兰州: 兰州理工大学, 2013