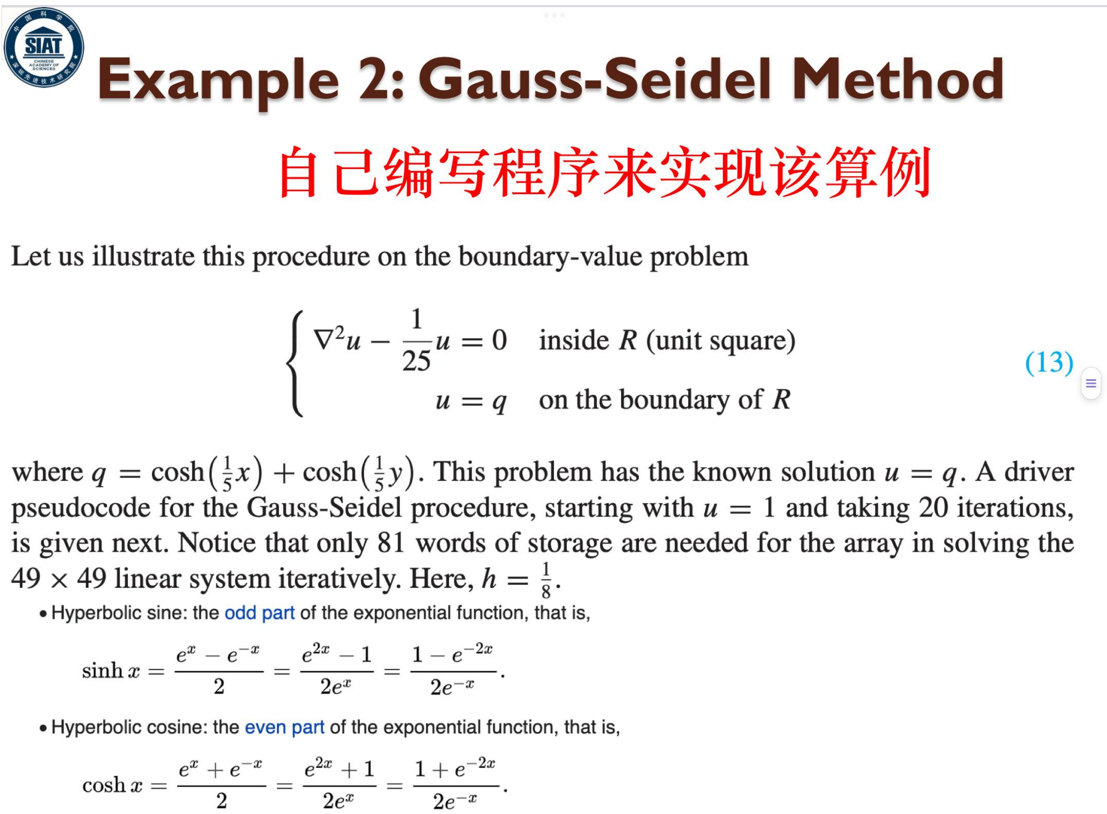
绘制 SIR 模型变化曲线

一、学生信息

姓名：柳建国
学号：2022Z8017782089
专业：电子信息
所部：数字所

二、问题描述

如图 1 所示，计算二维 Poisson 方程。



Example 2: Gauss-Seidel Method

自己编写程序来实现该算例

Let us illustrate this procedure on the boundary-value problem

$$\begin{cases} \nabla^2 u - \dfrac{1}{25} u = 0 & \text{inside } R \text{ (unit square)} \\ u = q & \text{on the boundary of } R \end{cases} \tag{13}$$

where $q = \cosh\left(\frac{1}{5}x\right) + \cosh\left(\frac{1}{5}y\right)$. This problem has the known solution $u = q$. A driver pseudocode for the Gauss-Seidel procedure, starting with $u = 1$ and taking 20 iterations, is given next. Notice that only 81 words of storage are needed for the array in solving the 49×49 linear system iteratively. Here, $h = \frac{1}{8}$.

- Hyperbolic sine: the odd part of the exponential function, that is,

$$\sinh x = \frac{e^x - e^{-x}}{2} = \frac{e^{2x} - 1}{2e^x} = \frac{1 - e^{-2x}}{2e^{-x}}.$$

- Hyperbolic cosine: the even part of the exponential function, that is,

$$\cosh x = \frac{e^x + e^{-x}}{2} = \frac{e^{2x} + 1}{2e^x} = \frac{1 + e^{-2x}}{2e^{-x}}.$$

图 1 Poisson 方程

三、实验过程

1. 实验原理

本次实验使用 Matlab 进行实现，相关代码见附录。

设网格数目为 m=n，网格步长为 h，在网格的四条边上，根据条件赋初始值。

根据题目构造离散化方程，可以得到：

$$\frac{\alpha^2 u}{\alpha x^2}+\frac{\alpha^2 u}{\alpha y^2}=\frac{1}{25}u$$

将上式二阶偏导数用有限差分表示，可得：

$$\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{h^2}+\frac{u_{i,j+1}-2u_{i,j}+u_{i,j-1}}{h^2}-\frac{1}{25}u_{i,j}=0$$

由此可以计算网格内的点。

2. 测试结果

当网格点数为 m=n=9 时，迭代 80 次，就可以达到最大误差小于 1e-6，绘制得到的图如图 2 所示。

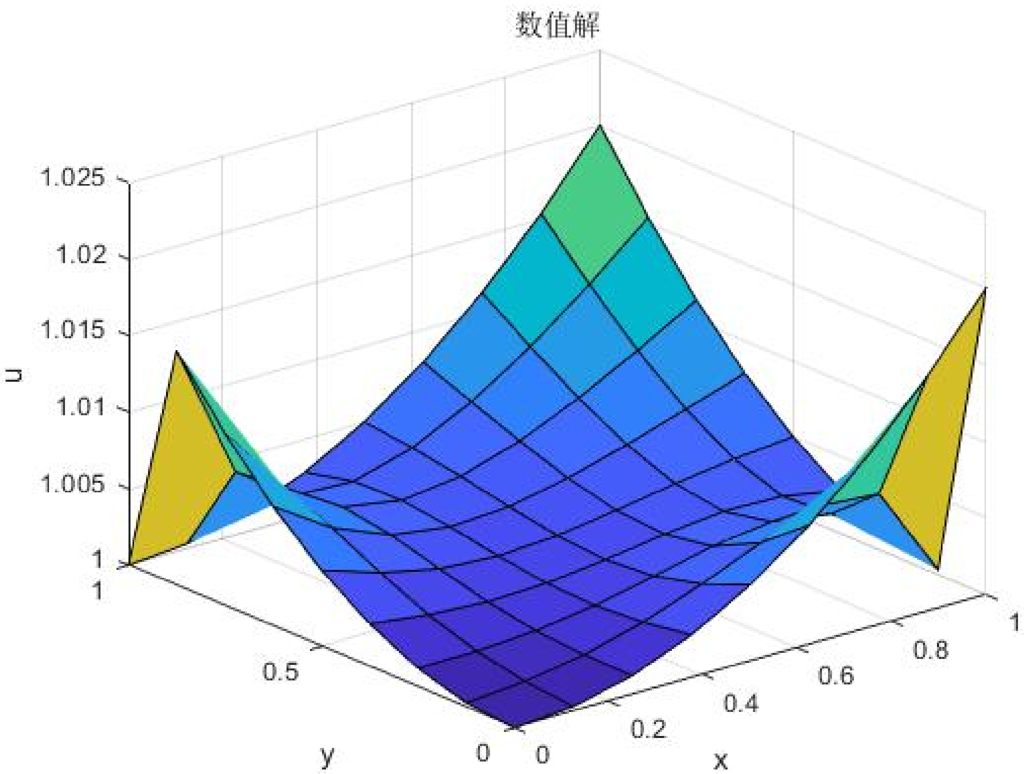


图 2 当网格为 m=n=9 时，误差小于 1e-6 时的结果

当网格点数为 m=n=100 时，迭代 1000 次后，最大误差为 0.0014，绘制得到的图形如图 3 所示。

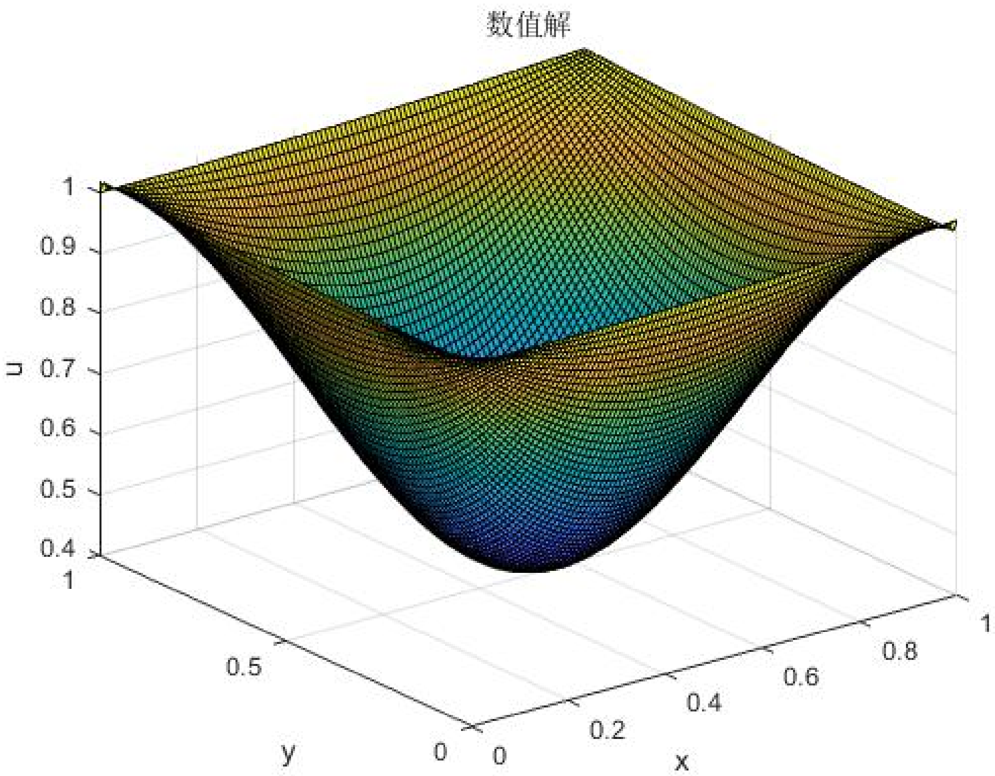


图 3 当网格为 m=n=100 时，迭代 100 次后，最大误差为 0.0014 的结果。

四、源代码-Matlab 实现

```
% 定义网格参数
m = 100; % 横向网格数
n = 100; % 纵向网格数
h = 1/(m-1); % 网格大小
x = 0:h:1;  % x 轴坐标
y = 0:h:1;  % y 轴坐标

% 初始化数值解
u = zeros(m,n); % 初始值为 0

% 设置边界条件
u(1,:) = cosh(1/5*x);  % 下边界温度
u(m,:) = cosh(1/5*x);  % 上边界温度
u(:,1) = cosh(1/5*y);  % 左边界温度
u(:,n) = cosh(1/5*y);  % 右边界温度

% 高斯-赛德尔迭代求解内部网格上的数值解
u0 = u;
max_iter = 1000;
```

```
tol = 1e-6;
for iter = 1:max_iter
    for i = 2:m-1
        for j = 2:n-1
            u(i,j) =
1/4*(u(i-1,j)+u(i+1,j)+u(i,j-1)+u(i,j+1)-h^2/25*u(i,j));
        end
    end
    % 检查相对误差是否小于容许度
    err = max(max(abs(u-u0)./abs(u)));
    if err < tol
        break;
    end
    % 更新迭代起点
    u0 = u;
end

% 可视化结果
[X,Y] = meshgrid(x,y);
surf(X,Y,u');
title('数值解');
xlabel('x');
ylabel('y');
zlabel('u');
```