

# SNM\_Report\_Project3

PB19111713钟颖康

## Program10

## 10.1 Introduction

### 程序 10 用 Newton 迭代法求解非线性方程组

$$\begin{cases} f(x) = x^2 + y^2 - 1 = 0 \\ g(x) = x^3 - y = 0 \end{cases}$$

取  $\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} 0.8 \\ 0.6 \end{pmatrix}$ , 误差控制  $\max(|x_k|, |y_k|) \leq 10^{-5}$ .

输入: 初始点  $(x_0, y_0) = (0.8, 0.6)$ , 精度控制值  $e$ , 定义函数  $f(x), g(x)$ .

输出: 迭代次数  $k$ , 第  $k$  步的迭代解  $(x_k, y_k)$ .

## 10.2 Method

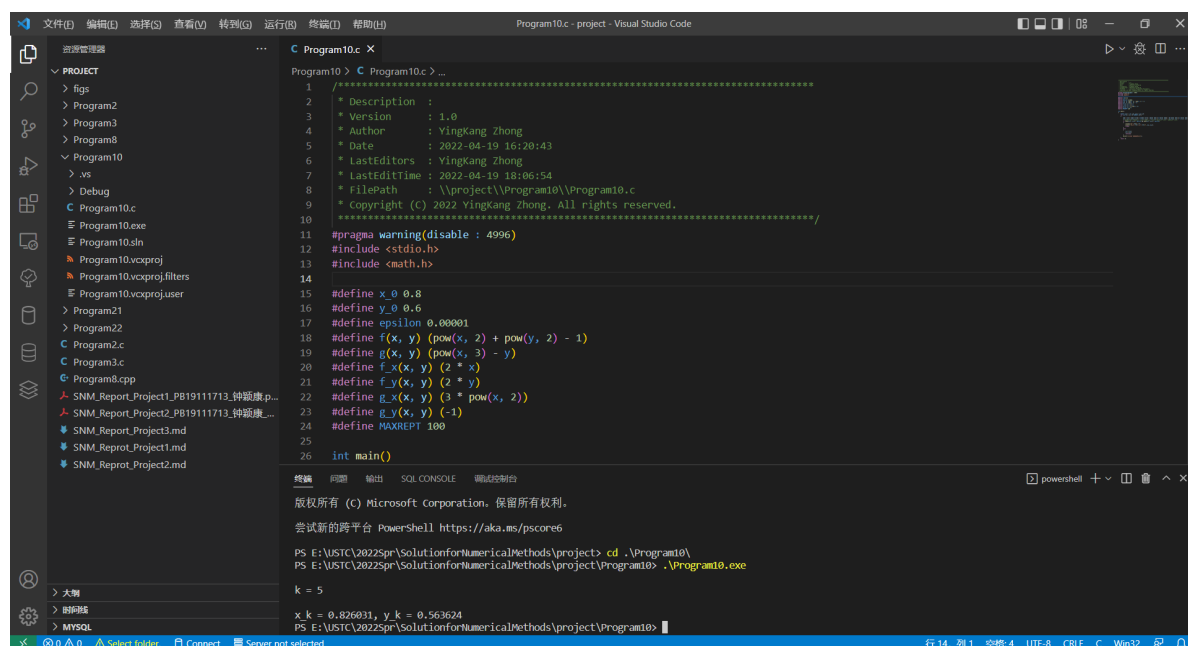
解课本P71的方程可得

$$\begin{cases} x_{k+1} = x_k + \frac{f(x_k, y_k) \cdot g_y(x_k, y_k) - g(x_k, y_k) \cdot f_y(x_k, y_k)}{g_x(x_k, y_k) \cdot f_y(x_k, y_k) - f_x(x_k, y_k) \cdot g_y(x_k, y_k)} \\ y_{k+1} = y_k + \frac{g(x_k, y_k) \cdot f_x(x_k, y_k) - f(x_k, y_k) \cdot g_x(x_k, y_k)}{g_x(x_k, y_k) \cdot f_y(x_k, y_k) - f_x(x_k, y_k) \cdot g_y(x_k, y_k)} \end{cases}$$

根据要求, 当 $\max(|x_k|, |y_k|) \leq 10^{-5}$ 时停止迭代。

据此可直接写出相应的C语言程序。

### 10.3 Result



根据程序运行结果，迭代到第5次时得到符合精度要求的迭代解

$$k = 5, \quad (x_k, y_k) = (0.826031, 0.563624)$$

## 10.4 Discussion

考虑到要实现从 *Input* 获取函数成本过于巨大，故直接在程序内部预设好函数。又由于C语言库中没有合适的求导函数，故对于预设的  $f(x)$ ,  $g(x)$  采取手动求偏导、手动定义的方式进行计算。

## Program21

### 21.1 Introduction

#### 程序 21 用二阶 Runge-Kutta 公式求解常微分方程组初值问题

$$\begin{cases} y'(x) = f(x, y), \\ y(a) = y_0, \end{cases} \quad a \leq x \leq b$$

$$(1) \begin{cases} y'(x) = y \sin \pi x, \\ y(0) = 1; \end{cases}$$

输入：区间剖分点数  $n$ , 区间端点  $a, b$ ; 定义函数  $y'(x) = f(x, y)$ .

输出： $y_k, k = 1, 2, \dots, n$ .

### 21.2 Method

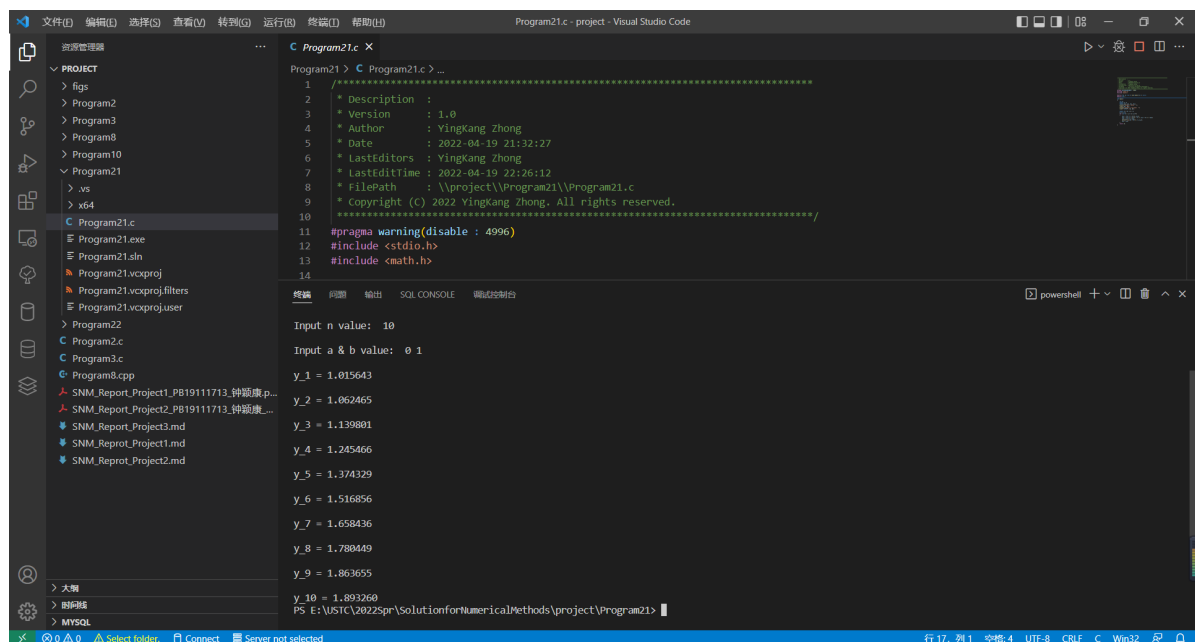
参考课本P148式(7.13)

$$\begin{cases} y_{n+1} = y_n + h(c_1 \cdot k_1 + c_2 \cdot k_2) \\ k_1 = f(x_n, y_n) \\ k_2 = f(x_n + a \cdot h, y_n + b \cdot h \cdot k_1) \end{cases}$$

简便起见，取  $c_1 = 0, c_2 = 1$ ，即采用中点公式进行编程计算：

$$\begin{cases} y_{n+1} = y_n + h \cdot k_2 \\ k_1 = f(x_n, y_n) \\ k_2 = f(x_n + a \cdot h, y_n + b \cdot h \cdot k_1) \end{cases}$$

### 21.3 Result



根据程序运行结果，对于 $n = 10, a = 0, b = 1$ ，有

k	y_k
1	1.015643
2	1.062465
3	1.139801
4	1.245466
5	1.374329
6	1.516856
7	1.658436
8	1.780449
9	1.863655
10	1.893260

## 21.4 Discussion

考虑到要实现从 $Input$ 获取函数成本过于巨大，故直接在程序内部预设好函数。经过搜索， $\pi$ 值决定采用反三角函数计算获得，即 $\text{acos}(-1)$ 。

## Program22

## 22.1 Introduction

使用程序22求解课本P156的例7.7

**程序 22** 用改进的 Euler 公式求解常微分方程组初值问题.

**计算公式:**

$$\begin{pmatrix} \bar{y}_{n+1} \\ \bar{z}_{n+1} \end{pmatrix} = \begin{pmatrix} y_n \\ z_n \end{pmatrix} + h \begin{pmatrix} f(x_n, y_n, z_n) \\ g(x_n, y_n, z_n) \end{pmatrix}$$
$$\begin{pmatrix} y_{n+1} \\ z_{n+1} \end{pmatrix} = \begin{pmatrix} y_n \\ z_n \end{pmatrix} + \frac{h}{2} \left[ \begin{pmatrix} f(x_n, y_n, z_n) \\ g(x_n, y_n, z_n) \end{pmatrix} + \begin{pmatrix} f(x_{n+1}, \bar{y}_{n+1}, \bar{z}_{n+1}) \\ g(x_{n+1}, \bar{y}_{n+1}, \bar{z}_{n+1}) \end{pmatrix} \right]$$

输入: 区间剖分点数  $N$ , 区间端点  $a, b$ ; 定义函数

$$y'(x) = f(x, y, z), \quad z'(x) = g(x, y, z)$$

输出:  $(y_k, z_k), k = 1, 2, \dots, N$ .

**例 7.7** 两种果树寄生虫, 其数量分别是  $u = u(t), v = v(t)$ , 其中一种寄生虫以吃另一种寄生虫为生, 两种寄生虫的增长函数如下列常微分方程组所示, 预测 3 年后这一对寄生虫的数量.

$$\begin{cases} \frac{du}{dt} = 0.09u \left(1 - \frac{u}{20}\right) - 0.45uv, \\ \frac{dv}{dt} = 0.06v \left(1 - \frac{v}{15}\right) - 0.001uv, \\ u(0) = 1.6, \\ v(0) = 1.2 \end{cases}$$

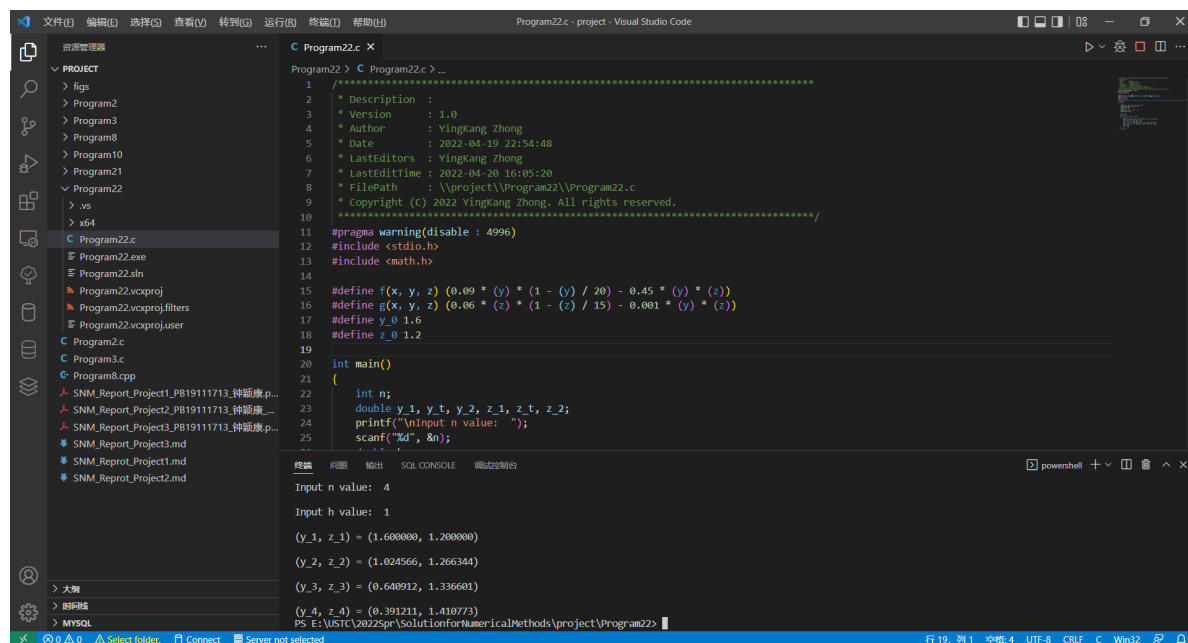
## 22.2 Method

## 计算公式:

$$\begin{pmatrix} \bar{y}_{n+1} \\ \bar{z}_{n+1} \end{pmatrix} = \begin{pmatrix} y_n \\ z_n \end{pmatrix} + h \begin{pmatrix} f(x_n, y_n, z_n) \\ g(x_n, y_n, z_n) \end{pmatrix}$$
$$\begin{pmatrix} y_{n+1} \\ z_{n+1} \end{pmatrix} = \begin{pmatrix} y_n \\ z_n \end{pmatrix} + \frac{h}{2} \left[ \begin{pmatrix} f(x_n, y_n, z_n) \\ g(x_n, y_n, z_n) \end{pmatrix} + \begin{pmatrix} f(x_{n+1}, \bar{y}_{n+1}, \bar{z}_{n+1}) \\ g(x_{n+1}, \bar{y}_{n+1}, \bar{z}_{n+1}) \end{pmatrix} \right]$$

根据计算公式可以直接写出对应的C语言程序。

## 22.3 Result



```
1 //*****
2 * Description :
3 * Version : 1.0
4 * Author : YingKang Zhong
5 * Date : 2022-04-19 22:54:48
6 * LastEditors : YingKang Zhong
7 * LastEditTime : 2022-04-20 16:05:20
8 * FilePath : \\project\\Program22\\Program22.c
9 * Copyright (C) 2022 YingKang Zhong. All rights reserved.
10 //*****
11 #pragma warning(disable : 4996)
12 #include <stdio.h>
13 #include <math.h>
14
15 #define f(x, y, z) (0.09 * (y) * (1 - (y) / 20) - 0.45 * (y) * (z))
16 #define g(x, y, z) (0.06 * (z) * (1 - (z) / 15) - 0.001 * (y) * (z))
17 #define y_0 1.6
18 #define z_0 1.2
19
20 int main()
21 {
22     int n;
23     double y_1, y_t, y_2, z_1, z_t, z_2;
24     printf("Input n value: ");
25     scanf("%d", &n);
26
27     Input n value: 4
28
29     Input h value: 1
30
31     (y_1, z_1) = (1.600000, 1.200000)
32
33     (y_2, z_2) = (1.024566, 1.266344)
34
35     (y_3, z_3) = (0.640912, 1.336601)
36
37     (y_4, z_4) = (0.391211, 1.410773)
38
39     PS E:\USTC\2022Spr\SolutioforNumericalMethods\project\Program22>
```

根据程序运行结果，

t	y_t	z_t
1	1.600000	1.200000
2	1.024566	1.266344
3	0.640912	1.336601
4	0.391211	1.410773

与课本上的结果比对，发现只有 $z_2$ 不一致，推测可能是课本印刷错误。