

doi: 10.3969/j.issn.1006-1576.2012.10.026

Excel 求解常微分方程组的两种方法

金晓龙

(广东女子职业技术学院应用设计系, 广州 511450)

摘要: 在实际应用中, 常需要建立多个物理量随某一物理量(如: 时间、温度等)变化的常微分方程组及对其求解, Excel 广泛应用于经济、金融、工程、科研、教学等领域, 用 Excel 求解常微分方程组具有一定的实际意义。笔者叙述 2 种 Excel 求解常微分方程组方法: 用 VBA 编程求解、调用 Matlab 求解, 并在实际编程环境中调试运行。

关键词: Excel; 常微分方程组; 数值解; Matlab

中图分类号: TP301.6 **文献标志码:** A

Two Excel Solutions of Ordinary Differential Equations

Jin Xiaolong

(Dept. of Application & Design, Guangdong Women's Polytechnic College, Guangzhou 511450, China)

Abstract: In practical applications, it often needs to establish the ordinary differential equations which physical quantities change over one quantity, such as time, temperature, etc. Excel is widely used in the fields of economy, finance, engineering, research, teaching and so on. It has some practical significance to solve the ordinary differential equations with Excel. This article describes two kinds of Excel method to solve ordinary differential equations, using VBA programming and calling Matlab. All programs run through the debugger and run well in the practical programming environment.

Key words: Excel; ordinary differential equations; numerical solution; Matlab

0 引言

微分方程是指, 方程中未知的是一个变量或几个变量的函数, 在方程中不仅有函数本身而且有它们的导数。如果方程中的未知函数是有多个变量的函数, 该方程为偏微分方程; 若未知函数是只有一个变量的函数, 该方程为常微分方程。微分方程中出现的未知函数导数的最高阶数, 为微分方程的阶数。在实际应用中, 经常需要建立若干个物理量随某一物理量(如: 时间、温度等)变化的常微分方程组及对其求解。在工程实际与科学研究中遇到的微分方程往往比较复杂, 在很多情况下都不适宜采用解析法来求解, 而需采用数值解法来求近似解。常微分方程数值解法的思路是: 对求解区间进行剖分, 然后把常微分方程离散成在节点上的近似公式或近似方程, 最后结合定解条件求出近似解。常微分方程组的数值解法有很多种, 在诸多算法中, 四阶龙格-库塔算法具有较高的精确度, 是一种优先选取的算法。

Excel 是微软的电子表格软件, 广泛应用于经济、金融、工程、科研、教学等领域, 使用 Excel 进行数据处理、绘制图表、分析决策等已成为人们日常工作内容之一。Excel 应用广泛, 而求解常微分方程组的数值解问题经常遇到, 所以, 用 Excel

对常微分方程组求解具有一定的实际意义。在 Excel 中常采用 2 种方法对常微分方程组求解: 一是使用 Excel 中的 VBA 编程求解; 二是用 VBA 调用 Matlab, 使用 Matlab 工具求解常微分方程组, 再将求解结果返回 VBA。第 1 种方法适合对具体常微分方程组求解, 并通过编程清晰认识求解过程。第 2 种方法适合对各种常微分方程组求解, 起到一种求解工具的作用。

1 常微分方程组求解的数学描述

常微分方程组与常微分方程解法不同之处在于用向量形式构成龙格-库塔公式。对一阶常微分方程组的初值问题

$$\begin{cases} y_i' = f_i(x, y_1, y_2, \dots, y_N) & (i=1, 2, \dots, N) \\ y_i(x_0) = y_i^0 & (i=1, 2, \dots, N) \end{cases}$$

采用向量描述方程组

$$\begin{aligned} \mathbf{y} &= (y_1, y_2, \dots, y_N)^T \\ \mathbf{y}^0 &= (y_1^0, y_2^0, \dots, y_N^0)^T \\ \mathbf{f} &= (f_1, f_2, \dots, f_N)^T \end{aligned}$$

则上述方程组的初值问题可表为

$$\begin{cases} \mathbf{y}' = \mathbf{f}(x, \mathbf{y}) \\ \mathbf{y}(x_0) = \mathbf{y}^0 \end{cases}$$

求解该方程组初值问题的四阶龙格-库塔公式为

收稿日期: 2012-05-02; 修回日期: 2012-06-14

作者简介: 金晓龙(1965—), 男, 上海人, 硕士, 副教授, 从事应用电子技术研究。

$$\begin{cases} y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 = f(x_n, y_n) \\ k_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1) \\ k_3 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2) \\ k_4 = f(x_n + h, y_n + hk_3) \end{cases}$$

高阶常微分方程(或方程组)可以化为一阶常微分方程组, 对下列 m 阶微分方程

$$y^{(m)} = f(x, y, y', \dots, y^{(m-1)})$$

初始条件为

$$y(x_0) = y_0, y'(x_0) = y'_0, \dots, y^{(m-1)}(x_0) = y_0^{(m-1)}$$

引进新的变量

$$y_1 = y, y_2 = y', \dots, y_m = y^{(m-1)}$$

可将 m 阶方程化为一阶方程组

$$\begin{cases} y_1' = y_2 \\ y_2' = y_3 \\ \dots \\ y_{m-1}' = y_m \\ y_m' = f(x, y_1, y_2, \dots, y_m) \end{cases}$$

初始条件相应转化为

$$y_1(x_0) = y_0, y_2(x_0) = y'_0, \dots, y_m(x_0) = y_0^{(m-1)}$$

2 VBA编程求解常微分方程组

Excel 包含 VBA 编程环境, 可通过编写程序处理 Excel 常用功能不能解决的问题, VBA 的语法与 VB 相同, 但 VB 程序可以独立运行, 而 VBA 程序需要借助宿主程序运行, 如在 Excel、Word 中运行。

以下面常微分方程组为例, 编程对其求解:

$$\begin{cases} y_1' = xy_1 + y_2 \\ y_2' = xy_2 \end{cases} \quad \text{在区间 } 0 \leq x \leq 1 \text{ 求解。}$$

$$\begin{cases} y_1(0) = 1 \\ y_2(0) = 1 \end{cases}$$

使用 VBA 编程(也可以其他语言), 采用四阶龙格-库塔公式对常微分方程组求解, 程序的流程图如图 1 所示。

编写宏函数。启动 Excel, “工具” → “宏” → “Visual Basic 编辑器”, 在资源管理器窗口, 右击某一图表, 选“插入” → “模块”。当创建将用在工作表中的自定义函数时, 要确保这些代码位于普通的 VBA 模块中, 如果将自定义函数放在 Sheet 或者 ThisWorkbook 的代码模块中, 那么它们在工

作表中就不能运行。下列代码实现四阶龙格-库塔运算, 由于是方程组, 需要 2 个解 y_1 、 y_2 , 所以在调用函数时使用了一个逻辑变量 kz , 为 True 返回 y_1 的值, 为 False 返回 y_2 的值。

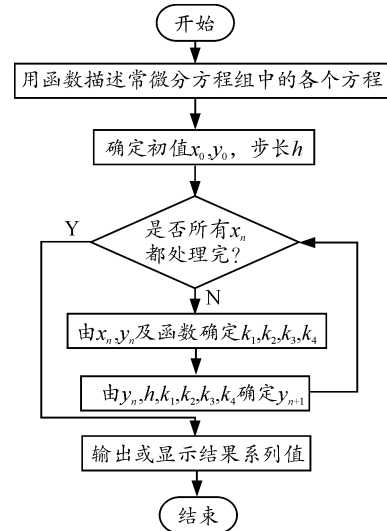


图 1 龙格-库塔算法流程图

代码中的 k_n 、 l_n 为向量 k_n 中成员。

```
Function f1(x, y1, y2)
    f1 = x*y1 + y2
End Function
Function f2(x, y1, y2)
    f2 = x*y2
End Function
Function fh(h, x, y1, y2, kz)
    k1 = f1(x, y1, y2)
    l1 = f2(x, y1, y2)
    k2 = f1(x + h/2, y1 + k1*h/2, y2 + l1*h/2)
    l2 = f2(x + h/2, y1 + k1*h/2, y2 + l1*h/2)
    k3 = f1(x + h/2, y1 + k2*h/2, y2 + l2*h/2)
    l3 = f2(x + h/2, y1 + k2*h/2, y2 + l2*h/2)
    k4 = f1(x+h, y1 + k3*h, y2 + l3*h)
    l4 = f2(x+h, y1 + k3*h, y2 + l3*h)
    If kz = True Then fh= y1 + (k1 + 2*k2 + 2*k3 + k4)
    *h/6
    If kz = False Then fh = y2 + (l1+2*l2+2*l3+l4)
    *h/6
End Function
```

在 Excel 单元格调用宏函数。用前一个状态的 x 、 y_1 、 y_2 、步长 h 来计算下一个状态的 y_1 、 y_2 , 在各单元格输入下列公式:

- A4单元格: =A3+\$B\$1
- B4单元格: =fh(\$B\$1,A3,B3,C3,True)
- C4单元格: =fh(\$B\$1,A3,B3,C3,False)

D4单元格: =(A4+1)×EXP(A4×A4/2)

E4单元格: =EXP(A4×A4/2)

选择 A4:E4 单元格区域, 鼠标移到右下角, 出现黑十字时, 拖动填充 A5:E13 单元格区域。运算结果如图 2 所示, 可以看到数值解与解析解非常接近。当下次打开该 Excel 文件时, 可能会禁止使用宏, 重新设置宏安全性, 选“工具”→“宏”→“安全性”→“中”, 保存文件, 关闭并重新打开文件。

	A	B	C	D	E
1	步长 h	0.1			
2	x	y ₁ 数值解	y ₂ 数值解	y ₁ 解析解	y ₂ 解析解
3	0	1	1	1	1
4	0.1	1.1055137719	1.0050125208	1.1055137729	1.0050125209
5	0.2	1.2242416033	1.0202013398	1.2242416080	1.0202013400
6	0.3	1.3598362051	1.0460278589	1.3598362179	1.0460278599
7	0.4	1.5166018649	1.0832870648	1.5166018947	1.0832870677
8	0.5	1.6997226137	1.1331484461	1.6997226796	1.1331484531
9	0.6	1.9155476414	1.1972173474	1.9155477810	1.1972173631
10	0.7	2.1719559482	1.2776212795	2.1719562324	1.2776213132
11	0.8	2.4788294186	1.3771276949	2.4788299758	1.3771277643
12	0.9	2.8486736953	1.4993023624	2.8486747501	1.4993025001
13	1	3.2974406045	1.6487210071	3.2974425414	1.6487212707
14					

图 2 VBA 编程运行结果

3 调用 Matlab 求解常微分方程组

Matlab 是美国 MathWorks 公司出品的商业数学软件, 用于算法开发、数据可视化、数据分析以及数值计算的高级计算语言和交互式环境。Matlab 中提供了一些采用龙格-库塔法来求解常微分方程的函数, 其中最常用的函数是 ode23(二三阶龙格-库塔函数)和 ode45(四五阶龙格-库塔函数)。函数的调用格式: [T,Y]=ODE45('F',TSPAN,Y₀), 输入参数中的'F'是一个字符串, 表示微分方程的形式, 也可以是 f(x,y)的 m 文件, 如果'F'为向量函数, 则用来解微分方程组, 'F'要写成变量的一阶微分表达式形式, 若为向量函数则用分号分隔; TSPAN=[T₀ TFINAL]表示积分区间; Y₀表示初始条件。

VBA 使用 ActiveX 调用 Matlab。在 VBA 中创建 Matlab 的 ActiveX 对象, 使用该对象的方法来实现对 Matlab 的调用, 该对象常用的 3 个方法: Execute、GetFullMatrix、PutFullMatrix。

1) Function Exectue (Name As String) as string。此函数用于执行一个合法的 Matlab 命令或者在 Matlab 的工作空间中调用 M 文件。

2) Sub GetFullMatrix(Name As String, workspace As String, pr() As Double), pi() As Double)。该方法将 Matlab 中的一个矩阵变量赋值到 VBA 的一个数组中, 其中: Name 是 Matlab 中矩阵变量名, Workspace 是该矩阵所在的 Matlab

工作区, pr 是 VBA 数组的实部, pi 是 VBA 数组的虚部。

3) Sub PutFullMatrix (Name As String, workspace As String, pr() As Double, pi() As Double)。该方法将 VBA 中的一个数组传送到 Matlab 中的一个矩阵变量中。

以前面的常微分方程组为例, 用 VBA 调用 Matlab 求解, 然后将结果返回 Excel 工作表中, Excel 用图形显示求解结果, 具体步骤如下:

选择“工具”下的“引用”标签, 选中 Matlab Application (Version 7.0) Type Library。将下列代码放入按钮单击事件中:

```

Dim Matlab As Object      '声明对象
Dim MReal1() As Double   '声明存放实部数组
Dim MImag() As Double    '声明存放虚部数组
Set Matlab = CreateObject("Matlab.Application")
Matlab.Execute ("fun=@(x,y) [x*y(1)+y(2);
x*y(2)];[x,y]=ode45(fun,[0:0.1:1], [1;1]);a=[x,y];")
'执行Matlab命令
Call Matlab.GetFullMatrix("a", "base", MReal1,
MImag)      '将结果分别存入实部、虚部数组
Set Matlab = Nothing
Sheets("Sheet3").Activate
For i=0 to 10 '将结果返回 Excel 工作表
Cells(3+i,1)=MReal1(i,0)
Cells(3+i ,2)=MReal1(i,1)
Cells(3+i ,3)=MReal1(i,2)
Next

```

利用 Excel 的绘图功能, 显示数值解的模拟曲线。选“插入”→“图表”→“XY 散点图”→“平滑线散点图”, 下一步, “数据区域”选 A3:C13, 选“系列产生在列”, 产生 2 个序列, 分别命名“y₁ 数值解”、“y₂ 数值解”, 下一步, 图表标题为“常微分方程组数值解图示”, 完成。双击 X 轴、Y 轴, 设置最大刻度及单位, 双击标题、图例等, 设置字体, 结果如图 3。

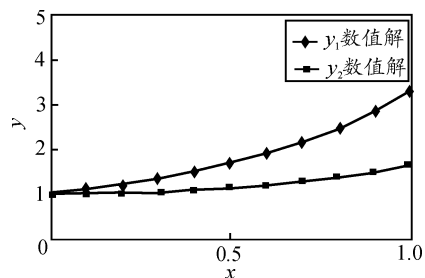


图 3 调用 Matlab 求解结果