

# Mathematica 软件在数值分析中的动态可视化设计

孔祥强

(菏泽学院 数学系, 山东 菏泽 274015)

**摘要:**利用 Mathematica 软件编程,实现了用复化 Simpson 公式求数值积分的可视化,动态演示了算例的计算过程;设计了用逐次超松弛迭代法求解线性方程组的交互过程,分析了松弛因子的改变对近似解的影响;给出了用一类特殊的牛顿迭代法求非线性方程根的可视化展示,通过算例演示了方法的求解过程。数值分析课程中利用 Mathematica 软件,可增强教学内容的直观性、交互性,激发学生学习的兴趣和动力。

**关键词:** Mathematica 软件;动态可视化;数值积分;线性方程组;非线性方程

中图分类号:O245 文献标志码:A 文章编号:1673-0143(2016)01-0074-07

DOI:10.16389/j.cnki.cn42-1737/n.2016.01.015

## Mathematica-Based Design of Dynamic Visualization in Numerical Analysis

KONG Xiangqiang

(Department of Mathematics, Heze University, Heze 274015, Shandong, China)

**Abstract:** Using the Mathematica software programming, the visualization of the compound Simpson formula for numerical integration is realized, and the dynamic calculation process of an example is presented. The interaction process to solve linear equations by the successive over-relaxation iteration method is designed, and the effect of the relaxation factor changes on the approximate solution is analyzed. A visual display of the special Newton iterative method for nonlinear equation roots is given, and the method of solving process through a numerical example is demonstrated. Using the Mathematica software in the course of numerical analysis, the intuitive and interactive of teaching content are enhanced, and the students' interests and motivation in learning are stimulated.

**Keywords:** Mathematica software; dynamic visualization; numerical integration; linear equations; nonlinear equation

Mathematica 软件是 Wolfram Research 公司 1988 年开发的一款科学计算类软件,具有集数值和符号计算、编程语言及图形绘制等为一身的强大功能。2013 年新推出了 Mathematica 9.0.1 版本,加强了在可视化方面的应用。在高等数学、微分方程和数学物理实验等大学数学课程中 Mathematica 软件得到了很好的应用<sup>[1-3]</sup>,通过提供的可视化函数可方便地编写程序<sup>[4-6]</sup>。但在数值分析课程中的应用比较有限,课程中比较繁琐的算法没有得到很好的可视化呈现。本文从动态交互可视化的角度出发,在 Mathematica 9.0.1 环境下,实现了算法的可视化操作,动态地展现了算法的计算过程。

数值分析课程包括数值积分、求线性方程组的解及求非线性方程的根等内容。① 求数值积分的方法有多种,复化 Simpson 公式是其中的一种,通过对算例的编程,设计了动态演示该公式求解的过程,达到可视化的效果;② 求线性方程组解的方法有雅克比迭代法、逐次超松弛迭代法(SOR)等,在首先判断

收稿日期:2015-07-19

基金项目:2013 年菏泽学院重点课题组项目(201311)

作者简介:孔祥强(1983—),男,讲师,硕士,研究方向:计算数学。

SOR 法收敛的情形下,通过选择不同的松弛因子,求出对应的近似解,分析松弛因子对方程组的解产生的影响;③ 非线性方程的数值解法有 Newton 迭代法等,通过对一类特殊的 Newton 迭代法的编程,以动画形式实现了求解过程,动态过程可手动操控,也可选用自动模式。

## 1 复化 Simpson 公式求数值积分的可视化实现

### 1.1 基本知识

设  $f(x)$  是闭区间  $[a, b]$  上具有连续光滑性质的函数,  $\rho(x)$  为开区间  $(a, b)$  内的权函数,利用数值方法求积分  $\int_a^b \rho(x)f(x) dx$ ,一般是找节点  $a \leq x_0 < \dots < x_n \leq b$  处  $f(x)$  对应函数值的某种线性组合来近似,即  $\int_a^b f(x) dx \approx \sum_{i=0}^n A_i f(x_i)$ ,其中积分系数  $A_i$  为独立于  $f(x)$  的常数,积分节点  $x_i, i=0, 1, \dots, n$ 。

(1) 构造基于插值多项式的插值型求积公式。

在  $[a, b]$  上,取  $\rho(x)=1, x_i=a+i\frac{b-a}{n}, (i=0, 1, \dots, n)$  的插值型公式称之为 Newton-Cotes 求积公式<sup>[7]</sup>。

设  $h=\frac{b-a}{n}, x=a+th$ , 则

$$A_k = \int_a^b l_k(x) dx = h \int_0^n \left( \prod_{\substack{i=0 \\ i \neq k}}^n \frac{t-i}{k-i} \right) dt = (b-a)C_k^{(n)}, C_k^{(n)} = \frac{(-1)^k}{n k! (n-k)!} \int_0^n \left( \prod_{\substack{i=0 \\ i \neq k}}^n (t-i) \right) dt, k=0, 1, \dots, n,$$

故 Newton-Cotes 求积公式为  $\int_a^b f(x) dx \approx (b-a) \sum_{k=0}^n C_k^{(n)} f(x_k)$ ,其中  $x_k=a+k\frac{b-a}{n}$ 。积分系数  $C_k^{(n)}$  与函数  $f(x)$  及区间  $[a, b]$  均无关,称为 Cotes 系数。Cotes 系数可通过计算或查表得到。特别地,称  $n=2$  时的 Newton-Cotes 公式为 Simpson 公式<sup>[7]</sup>,即

$$\int_a^b f(x) dx \approx \frac{b-a}{6} [f(a)+f(b)+4f(\frac{a+b}{2})]。$$

(2) 构造基于正交多项式的 Gauss 型求积公式。

若  $a < x_0 < \dots < x_n < b$  是  $[a, b]$  上关于权函数  $\rho(x)$  的  $n+1$  次正交多项式  $P_{n+1}(x)$  的根,则称具有  $n+1$  个节点的插值型公式  $\int_a^b \rho(x)f(x) dx \approx \sum_{k=0}^n A_k f(x_k)$  为 Gauss 型求积公式<sup>[8]</sup>。

无论是 Newton-Cotes 型求积公式还是 Gauss 型求积公式,提高代数精度的重要方法是增加节点的个数,但随着个数的增加,Newton-Cotes 求积公式的稳定性变差,Gauss 型求积公式的积分系数一般为无理数,计算十分不便,此时可选择另一种提高精度的方法,即复化求积公式法。等距细分区间  $[a, b]$ ,  $x_i=a+i\frac{b-a}{n}, h=\frac{b-a}{n}, i=0, 1, \dots, n$ , 在小区间  $[x_{i-1}, x_i]$  上采用 Simpson 公式,可得复化 Simpson 求积公式<sup>[7]</sup>

$$\int_a^b f(x) dx \approx \frac{h}{6} [f(a)+f(b)+2 \sum_{k=1}^{n-1} f(x_{k-1})+4 \sum_{k=1}^n f(x_{k-\frac{1}{2}})]。$$

### 1.2 可视化设计

以计算积分  $\int_0^3 [1-e^{-x} \sin 3x] dx$  为例,求得具有 10 位精度的真实值为 2.688 443 066,利用复化 Simpson 公式求出具有 10 位精度的近似值,当选取的区间个数不同时,相应的近似值实时变化;真实值减去近似值的绝对值为误差值,随近似值的变化,误差值实时改变。选取的区间通过不同的颜色进行填充,填充部分的面积即为近似值。实现的主要功能体现在界面图 1 中。界面左下角的控件可以手动拖动滑块,也可以点击“+显示动画控制”按钮<sup>[9]</sup>,实现自动播放的功能,可以调节播放的快慢、前进和后退等。

### 1.3 编程实现

Manipulate[

a=0; b=3; n=t; h=(b-a)/(2n);

```

f[x_]:=1-Exp[-x]*Sin[3x];
zs=N[abrx,10];
js=N[1/3*h*(f[a]+f[b]+2n1sk1f[a+2k h]+4nsk1f[a+2k 1 h]),10];
w=Abs[js-zs];
Q=Table[{a+i h ,f[a+i h]}, {i,0,2n,1}];
For[i=1;m=1,i≤2n,i=i+2;m++, u[m]=Q[[i;;i+2]];
  fig[m]=Plot[Evaluate[Fit[u[m],{1,x,x^2},{x}]]
    ,{x,(m-1)2h,2h*m}
    ,PlotRange→{{0,3.1},{0,3.1}}
    ,FillingStyle→Hue[0.2m+0.051,1,1,0.6]
    ,Filling→Axis];];
d1=Plot[f[x],{x,0,3}
  ,PlotStyle→Opacity[0.9]
  ,PlotRange→{{-0.1,3.3},{0,2}}
  ,Epilog→{Text[Style[Grid[{
    {Style["真实值",Blue,Bold,15],zs}
    ,{Style["近似值",Blue,Bold,15],js}
    ,{Style["误差值",Blue,Bold,15],w}
    ,{Style["区间个数",Blue,Bold,15],t}}
    ,Frame→All],17},{2.4,1.7}]}];
d2=ListPlot[Q,PlotStyle {Hue[0.2t],PointSize[0.025]}];
Show[{d1,d2,Table[fig[s],{s,n}]}
  ,ImageSize→600,AxesStyle→Arrowheads[0.02]]
,{t,1,Style["区间个数",Red,Bold,15]},1,6,1
  ,Appearance→"Labeled"}
  ,ControlPlacement→Bottom]

```

在上面的程序中,首先定义出函数  $f(x)$ ,若取  $h = \frac{b-a}{2n}$ ,则复化 Simpson 求积公式变为  $\int_a^b f(x) dx \approx \frac{h}{3}[f(a)+f(b)+2\sum_{k=1}^{n-1}f(x_{k-1})+4\sum_{k=1}^n f(x_{k-\frac{1}{2}})]$ 。  $n=4$  时,真实值  $zs=2.688443066$ ,近似值  $js=2.686397838$ ,误差值  $w=0.002045228$ ;利用 Epilog 函数绘出界面右上方的图形基元列表;Plot 函数绘出函数  $f(x)$  的图形及带填充色彩的图形,Show 函数可将图形同时显示出来;最后,通过可视化函数 Manipulate 实现动态交互的可视化功能。计算其他函数的数值积分可利用同一程序实现。

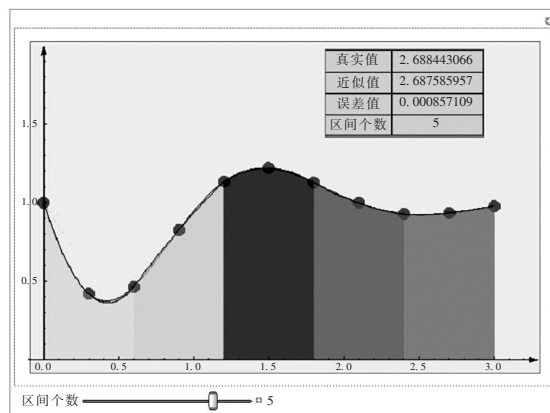


图1 复化 Simpson 公式的动态模拟

Fig. 1 Dynamic simulation of complex Simpson formula

## 2 逐次超松弛迭代法(SOR)解线性方程组的可视化实现

### 2.1 基本知识

线性方程组是很多数学分支都涉及到的问题<sup>[10-11]</sup>。设  $A \in \mathbf{R}^{n \times n}$  且  $|A| \neq 0$ ,  $Ax = b$ , 对  $A$  进行如下的分解:

$$A = \begin{pmatrix} a_{11} & & & \\ & a_{22} & & \\ & & \ddots & \\ & & & a_{nn} \end{pmatrix} - \begin{pmatrix} 0 & & & \\ -a_{21} & 0 & & \\ \vdots & \vdots & \ddots & \\ -a_{n1} & \cdots & -a_{n,n-1} & 0 \end{pmatrix} - \begin{pmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ & 0 & & \\ & & \ddots & \\ & & & -a_{n-1,n} \\ & & & & 0 \end{pmatrix} \equiv D - L - U \quad (a_{ii} \neq 0).$$

现将  $A$  分裂为  $A = M - N$ , 其中  $M$  一般选为  $A$  的某种近似矩阵, 且  $|M| \neq 0$ ,  $Mx = d$  易求解。于是  $Ax = b \Leftrightarrow Mx = Nx + b$ , 或  $x = M^{-1}Nx + M^{-1}b$ , 从而构造出一阶定常迭代法为:

$$\begin{cases} x^{(0)} & \text{初始向量,} \\ x^{(k+1)} = Bx^{(k)} + f & k = 0, 1, \dots, \end{cases}$$

其中  $f = M^{-1}b$ ,  $B = M^{-1}N = M^{-1}(M - A) = I - M^{-1}A$ ,  $B$  称为迭代矩阵。若取矩阵  $M = \frac{1}{\omega}(D - \omega L)$  (带参数的下三角矩阵),  $\omega > 0$  为可选择的松弛因子,  $A = M - N$ , 可构造出逐次超松弛迭代法(SOR)的迭代格式:

$$\begin{cases} x^{(0)} & \text{初始向量,} \\ x^{(k+1)} = L_{\omega}x^{(k)} + f, \end{cases}$$

其中  $f = \omega(D - \omega L)^{-1}b$ ,  $L_{\omega} = (D - \omega L)^{-1}((1 - \omega)D + \omega U)$ ,  $L_{\omega} = (D - \omega L)^{-1}((1 - \omega)D + \omega U)$  为迭代矩阵<sup>[7]</sup>。

迭代法基本定理<sup>[7]</sup>: (1) 设方程组  $x = Bx + f$ ,  $B \in \mathbf{R}^{n \times n}$ , (2) 迭代法  $x^{(k+1)} = Bx^{(k)} + f$ , 对任意的初始向量  $x^{(0)}$ , 迭代法收敛的充要条件是  $B$  的所有特征值  $\lambda_i(B)$  满足  $|\lambda_i(B)| < 1$  或谱半径  $\rho(B) < 1$ 。若解  $Ax = b$  的 SOR 法收敛, 则  $0 < \omega < 2$ ; 当  $\omega \leq 0$  或  $\omega \geq 2$  时, SOR 法不收敛。

### 2.2 可视化设计

以求解线性方程组  $\begin{pmatrix} 5 & 2 & 1 \\ -1 & 4 & 2 \\ 2 & -3 & 10 \end{pmatrix} x = \begin{pmatrix} -12 \\ 20 \\ 3 \end{pmatrix}$  为例, 其精确解  $x = \{-4, 3, 2\}$ , 实现了当选取不同的松弛因子  $\omega$  时, 所得迭代矩阵  $L_{\omega}$  随之变化, 谱半径、迭代次数及近似解也实时改变, 其主要功能显示在图 2 中。

松弛因子  $\omega \in [0.01, 1.43] \subset (0, 2)$ , 在该范围内, 迭代矩阵的谱半径  $\rho(L_{\omega}) < 1$ , 保证了 SOR 方法的收敛性。通过滑动界面左下方的滑块可手动调节松弛因子, 也可点击“+显示动画控制”按钮<sup>[9]</sup>, 实现自动播放的功能, 同时可以调节播放的快慢、前进和后退等。

松弛因子  $\omega \in [0.01, 1.43] \subset (0, 2)$ , 在该范围内, 迭代矩阵的谱半径  $\rho(L_{\omega}) < 1$ , 保证了 SOR 方法的收敛性。通过滑动界面左下方的滑块可手动调节松弛因子, 也可点击“+显示动画控制”按钮<sup>[9]</sup>, 实现自动播放的功能, 同时可以调节播放的快慢、前进和后退等。

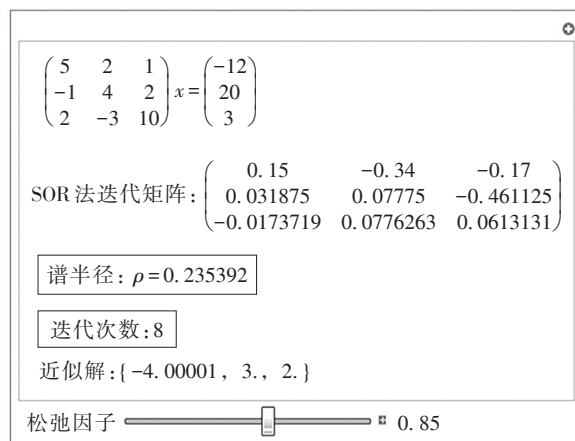


图2 SOR方法求解方程组的动态模拟

Fig. 2 Dynamic simulation of SOR method for solving equations

### 2.3 编程实现

```

Manipulate[
  A={{5, 2, 1}, {-1, 4, 2}, {2, -3, 10}};
  b={-12, 20, 3}; T=300; w=p; epi=0.0001;
  B=Table[A[[i, i]], {i, 1, 3}];
  B=DiagonalMatrix[B]; h=Inverse[B].b;
  B=IdentityMatrix[3]-Inverse[B].A;
  U=Table[If[i<j, B[[i, j]], 0], {i, 3}, {j, 3}];
  L=Table[If[i>j, B[[i, j]], 0], {i, 3}, {j, 3}];
  Lw=Inverse[IdentityMatrix[3]-w*L].(w*U+(1-w)*IdentityMatrix[3]);
  ρ =Max[Abs[Eigenvalues[Lw]]];
  m=w*Inverse[IdentityMatrix[3]-w*L].h;
  X0=Table[1, {i, 1, 3}];
  For[i=1, 3, i++, X=Lw.X0+m/N;
    If[Apply[Plus, Sqrt[Abs[(X-X0)^2]]] ≤ epi, Break[]];
    t=i; X0=X; ];
  Column[{
    Style[Row[{ MatrixForm[A], "X=", MatrixForm[b] }], 15], ,
    , Style[Row[{Style["SOR 法迭代矩阵:", Blue, Bold, 15], MatrixForm[Lw]}],
      15],
    , Framed[Style[Row[{ Style["谱半径: ρ =", Blue, Bold, 15], ρ }], 15]],
    , Framed[Style[Row[{Style["迭代次数:", Blue, Bold, 15], t}], 14]]
    , Style[Row[{ Style["近似解:", Blue, Bold, 15], N[X] }], 15]]
    , {{p, 0.85, Row[{Style[" 松弛因子", Red, Bold, 15]}]}, 0.01, 1.43, 0.02
      , Appearance→"Labeled"}
    , ControlPlacement→Bottom ]

```

在上面的程序中,若取  $\omega=0.93$ ,则得谱半径  $\rho(L_\omega)=0.184655$ ,迭代次数为7次,近似解为  $\{-4, 3, 2\}$ ,和精确解一致,无误差;迭代矩阵

$$L_\omega = \begin{pmatrix} 0.07 & -0.372 & -0.186 \\ 0.016275 & -0.01649 & -0.508245 \\ -0.00847927 & 0.0645913 & -0.0372044 \end{pmatrix}。$$

对于其他类似的线性方程组的求解也可以通过同一程序实现。

## 3 一类特殊牛顿迭代法的可视化实现

### 3.1 基本知识

对非线性方程  $f(x)=0$ ,设  $f(x)$  二次连续可导,  $x_k$  为  $f(x)=0$  的第  $k$  次近似解,用过点  $(x_k, f(x_k))$  的切线  $y=f(x_k)+f'(x_k)(x-x_k)$  近似代替曲线  $y=f(x)$ ,取切线与  $x$  轴的交点为  $f(x)=0$  的第  $k+1$  近似解,  $x_{k+1}=x_k-f(x_k)/f'(x_k)$ ,即为牛顿迭代法公式<sup>[7]</sup>。牛顿迭代公式是用点  $(x_k, f(x_k))$  处的切线与  $x$  轴的交点近似代替曲线  $y=f(x)$  与  $x$  轴的交点。在公式中,要计算导数  $f'(x)$ ,当  $f'(x)$  不易计算时,通常用初始点的导数  $f'(x_0)$  近似代替  $f'(x_k)$ ,得一类特殊的牛顿迭代法公式:  $x_{k+1}=x_k-f(x_k)/f'(x_0)$ ,此时的迭代序列  $\{x_k\}$  是一系列平行直线与  $x$  轴的交点<sup>[7]</sup>。

### 3.2 可视化设计

以求非线性方程  $8x^2-\sin x-0.25=0$  的根为例,初值  $x_0=0.61$ ,在要求计算精度为  $10^{-6}$  的条件下,动态观察求解的过程,主要功能体现在图3中。

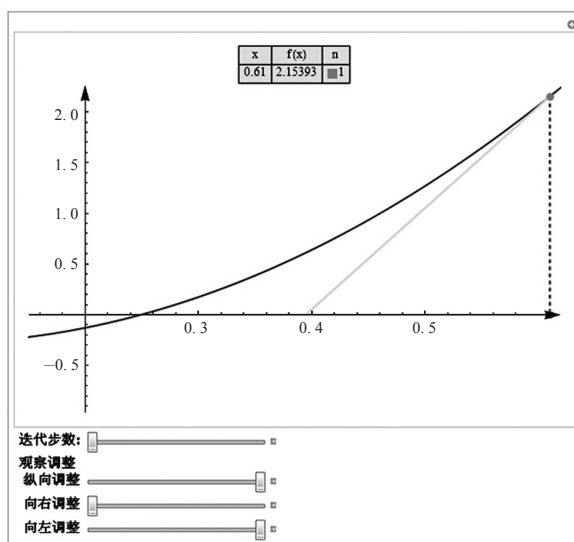


图3 特殊牛顿迭代法求非线性方程根的动态模拟

Fig. 3 Dynamic simulation of special Newton iteration for solving roots of nonlinear equation

随着迭代步数的增加,平行直线与  $x$  轴的交点逐步靠近曲线  $y=f(x)$  与  $x$  轴的交点(方程的根),在满足精度的条件下,可得方程的根。界面上方的表格内,  $n$  为迭代次数,  $x$  为迭代点的值,迭代点对应的函数值为  $f(x)$ , 这些值均实时变化。界面左下方,设置观察调整按钮,可纵向调整、向右调整和向左调整,便于调整观察角度;也可点击右方“+显示动画控制”按钮<sup>[9]</sup>,实现动画的自动播放。

### 3.3 编程实现

```

Manipulate[
  f[x_]:=8*x^2-Sin[x]-0.25;
  cishu=150;wucha=0.000001;jingdu=0.000001;
  x0=0.61; z=10; x2={0.61};
  For[j=1, j<cishu, j++,
    x1=x0-f[x0]/z;
    x2=Append[x2, x1];
    If[Abs[x0-x1]>jingdu,
      x0=x1, Break[]];];
  p=Plot[f[x], {x, 0.15, 0.65}
    , PlotStyle->Thickness[0.003]
    , PlotLabel->Style[
      Grid[{"x", "f(x)", "n"}
        , {x2[[t]], f[x2[[t]]}, Row[{Graphics[{Hue[0.1t]
          , Rectangle[]}, ImageSize->13], t}]}
    ], Frame->All, Background->LightBlue], 14]
    , AxesStyle->Arrowheads[0.03]];
  fig=Table[Graphics[{
    {Thickness[0.004], Hue[0.15j], Line[{x2[[j]]
      , f[x2[[j]]}], {x2[[j+1]], 0}}},
    {Red, Dashed, Thickness[0.004], Line[{x2[[j]]
      , f[x2[[j]]}], {x2[[j]], 0}}},
    {Hue[0.1j], PointSize[0.015], Point[{x2[[j]], 0}]},
    {Hue[0.1j], PointSize[0.015], Point[{x2[[j]], f[x2[[j]]}]}]}

```

```

    }], {j, 1, 15}];
Show[ {p, Table[ fig[ [s] ], {s, 1, t} ] ]
    , ImageSize → 600, PlotRange → {{c, d}, {-0.9, e}} ]
, {t, 1, Style[ "迭代步数:", Blue, Bold, 15 ] }, 1, 15, 1]
, Style[ "观察调整", Blue, Bold, 15 ]
, {e, 2.2, Style[ "纵向调整", Blue, Bold, 15 ] }, 1, 2.2, 0.1]
, {c, 0.16, Style[ "向右调整", Blue, Bold, 15 ] }, 0.16, 0.61, 0.04]
, {d, 0.61, Style[ "向左调整", Blue, Bold, 15 ] }, 0.16, 0.61, 0.04]
, ControlPlacement → Bottom ]

```

上面的程序中,若迭代步数  $n=13$ ,此时得  $x=0.250882$ ,  $f(x)=0.0052763$ ,求出了满足精度的方程的根。程序中的方程  $8x^2 - \sin x - 0.25 = 0$  也可以换成其他的类型,或者设计具有下拉菜单功能的控件,通过选择不同的方程,动态演示每个方程求根的可视化过程,达到人机交互的目的。

#### 4 结语

将完成的 Mathematica Notebook 文档保存为.nb 格式,便可在 Mathematica 9.0.1 环境下运行;程序中涉及的图形可利用函数 Export 将其输出,图形以.gif 格式默认保存在“我的文档”中。

文中的动态可视化研究方法对数值分析课程中的其他算法也是适合的,如雅克比迭代法、G-S 迭代法等都可以仿照文中编程实现其动态功能效果,Mathematica 软件提供的强大动态可视化功能比其他数学软件无法比拟的。引入数学软件进课堂,可实现数学知识的直观性、互动性和交互性<sup>[12]</sup>,对提高教学质量和激发学生学习的兴趣是大有裨益的。

#### 参考文献 (References)

- [1] 马千里. Mathematica 软件在微积分教学中的应用[J]. 陕西师范大学学报(自然科学版), 2009(S1): 135-136.
- [2] 秦大康, 张宝善. Mathematica 软件在微分方程中的应用研究[J]. 徐州师范大学学报(自然科学版), 2001, 19(3): 9-11.
- [3] 金哲, 张凤兰. Mathematica 与实验数据分析[J]. 延边大学学报(自然科学版), 2003, 29(1): 36-39.
- [4] RUSKEPÄÄ H. Mathematica navigator: mathematics, statistics, and graphics[M]. 3rd ed. San Diego: Elsevier, 2009.
- [5] 吴飞. Mathematica 项目演示笔记[M]. 北京: 清华大学出版社, 2010.
- [6] MANGANO S. Mathematica cookbook[M]. New York: O'Reilly Media, 2010.
- [7] 易大义, 陈道琦. 数值分析引论[M]. 杭州: 浙江大学出版社, 2002: 162-372.
- [8] 吴宗敏, 苏仰峰. 数值逼近[M]. 北京: 科学出版社, 2008: 51-56.
- [9] 郑永凡, 王艳青. 基于 Mathematica 的交互动态可视化设计及其应用[J]. 辽宁大学学报(自然科学版), 2010, 37(4): 324-328.
- [10] LIPSCHUTZ S. Schaum's outline of theory and problems of linear algebra[M]. New York: McGraw-Hill, 1968.
- [11] 王福贵, 侯建文. 基于 Mathematica 的线性方程组的可视化求解[J]. 计算机与现代化, 2013(5): 149-154.
- [12] 王同科, 张东丽, 王彩华. Mathematica 与数值分析实验[M]. 北京: 清华大学出版社, 2011.

(责任编辑: 范建凤)