

湖南农业大学

全日制普通本科生毕业论文

时滞微分方程多种数值解法的计算机实现

**SOLVING TIME-DELAY DIFFERENTIAL EQUATION IN
SEVERAL NUMERICAL WAYS WITH COMPUTER**

学生姓名：熊君武

学 号：200340204219

年级专业：2003 级信息与计算科学 2 班

指导老师：周铁军 教授

学 院：理学院

湖南·长沙

提交日期：2007年5月

湖南农业大学全日制普通本科生毕业论文（设计）

诚信声明

本人郑重声明：所呈交的本科毕业论文（设计）是本人在指导老师的指导下，进行研究工作所取得的成果，成果不存在知识产权争议。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体在文中均作了明确的说明并表示了谢意。本人完全意识到本声明的法律结果由本人承担。

毕业论文（设计）作者签名：

年 月 日

目 录

摘要	3
关键词	3
1 前言	4
1.1 选题背景综述	4
1.1.1 常微分方程的数值解法	4
1.1.2 双向联想记忆神经网络模型	4
2 时滞微分方程各种数值解算法介绍	6
2.1 欧拉法算法描述	6
2.2 改进欧拉法算法描述	7
2.3 龙格-库塔法算法描述	9
2.4 半离散法算法描述	11
3 时滞微分方程多种数值解法的计算机实现及其可视化处理	14
3.1 欧拉法、改进欧拉法实例与计算机仿真	14
3.2 龙格-库塔法算法实例与计算机仿真	16
3.3 半离散法算法实例与计算机仿真	16
4 时滞微分方程解各种数值解法性能对比分析	18
4.1 欧拉法、改进欧拉法和龙格-库塔法性能分析	18
4.2 半离散算法性能分析	19
5 结论	21
参考文献	22
致 谢	23

时滞微分方程多种数值解法的计算机实现

作者：熊君武

指导老师：周铁军

(湖南农业大学理学院，长沙 410128)

摘要：求解常微分方程数值解的方法有欧拉法、改进的欧拉法、龙格-库塔法以及半离散法等，这些方法在进行计算时性能上是有差异的，本文以双向联想神经网络模型为例，对模型的时滞差分方程运用不同数值解法求解，对其解的性能进行对比研究，并用 MatLab 在计算机实现这些解法，进行计算机仿真，通过不同方法的数值仿真，对各种解的性能进行可视化处理，对比它们的性能，最后根据仿真图形对这些时滞微分方程的平衡点、周期解的存在性与稳定性进行简单地讨论。

关键词：时滞微分方程；欧拉法；改进的欧拉法；龙格-库塔法；半离散法；双向联想记忆神经网络

Solving Time-delay Differential Equations in Several Numerical Algorithms with Computer

Student: XIONG, Jun-wu

Tutor: ZHOU, Tie-jun

(College of science, Hunan Agricultural University, Changsha 410128, China)

Abstract: There is different numerical algorithms solving ordinary differential equations, such as Euler, Improved Euler, Runge - Kutta and semi - discrete method, which vary greatly in computational performance. In this essay, we take bi-directory associative memory neutral network model for example, solve the time-delay differential equations of the model in several numerical ways and compare their different computational performance with MatLab. In addition, I briefly study their equilibrium point, the existence of periodic solutions and stability according to the result of simulation.

Key words: Time-Delay differential equations; Euler; Improved Euler; Runge - Kutta method; Semi - discrete; BAM

1 前言

1.1 选题背景综述

1.1.1 常微分方程的数值解法

常微分方程的概念、解法和理论很多，比如方程和方程组的种类及解法、解的存在性和唯一性、奇解、定性理论等。求通解在历史上曾作为微分方程的主要目标，求出通解的表达式后就容易从中得到问题所需要的特解。此外，也可以由通解的表达式，了解对某些参数的依赖情况，便于参数取值适宜，使它对应的解具有所需要的性能，还有助于进行关于解的其他研究。后来的发展表明，能够求出通解的情况不多，在实际应用中所需要的多是求满足某种指定条件的特解。当然，通解是有助于研究解的属性的，但是人们已把研究重点转移到定解问题上来。一个常微分方程是不是有特解呢？如果有，又有几个呢？这是微分方程论中一个基本的问题，数学家把它归纳成基本定理，叫做存在和唯一性定理。因为如果没有解，而我们要去求解，那是没有意义的；如果有解而又不是唯一的，那又不好确定。因此，存在和唯一性定理对于微分方程的求解十分重要。但是，即使存在特解，也只有极少数的微分方程类型可以用初等函数显式地写出精确解析解，工程技术及自然科学各部门的需要使得人们去寻求近似解析解，即解析形式的容许一定误差的数值解^[5]，即通过加减乘除和函数运算，求得特定点的近似解。本文以双向联想记忆神经网络模型为例，借助于神经网络模型建立的时滞差分方程，运用不同数值解法对其解的性能进行对比研究^[16]，利用 Mat1Lab 在计算机实现多种数值解法并其绘制图形^[14]，对各种解的性能进行可视化仿真处理，而且根据仿真结果对一些时滞微分方程研究其平衡点、周期解的存在性与稳定性，通过不同方法的数值仿真，对比它们的性能。这使得我们能够借助数学的方法，利用神经网络理论的研究成果，更好地研究神经网络算法的稳定性、收敛性、周期性、容错性、鲁棒性等性能，探索功能更加完善、性能更加优越的神经网络模型^[15]。

1.1.2 双向联想记忆神经网络模型^[9]

本文以双向联想神经网络模型为例，对以下几个模型的时滞差分方程运用不同数值解法求解，分别叙述如下^[1]：

(1) BAM 神经网络无时滞微分方程模型

Kosko 研究的一类无时滞 BAM 神经网络具有以下形式^[2]：

$$\begin{cases} \frac{dx_i(t)}{dt} = -a_i x_i(t) + \sum_{j=1}^p w_{ji} f_j(y_j(t)) + I_i, \\ \frac{dy_j(t)}{dt} = -b_j y_j(t) + \sum_{i=1}^m v_{ij} g_i(x_i(t)) + J_j, \end{cases} \quad (1)$$

($i = 1, 2, \dots, m$, $j = 1, 2, \dots, p$), 其中 $x_i(t)$ 表示域 1 中第 i 个神经元在时刻 t 的膜电压, 而 $y_j(t)$ 表示域 2 中第 j 个神经元在时刻 t 的膜电压, $a_i > 0$ 、 $b_j > 0$ 分别表示域中神经元的信号衰减系数, $g_i(\cdot)$ 、 $f_j(\cdot)$ 分别表示域 1 与域 2 中神经元的信号传递函数, w_{ji} 表示域 2 中第 j 个神经元与域 1 中第 i 个神经元的联接强度, v_{ij} 表示域 1 中第 i 个神经元与域 2 中第 j 个神经元的联接强度, I_i 是域 1 中第 i 个神经元的外部输入, J_j 是域 2 中第 j 个神经元的外部输入。

(2) BAM 神经网络常数时滞微分方程模型

Gopalsamy 和 He 研究的一类常数时滞 BAM 神经网络形式如下^[3]:

$$\begin{cases} \frac{dx_i(t)}{dt} = -a_i x_i(t) + \sum_{j=1}^p w_{ji} f_j(y_j(t - \sigma_{ji})) + I_i, \\ \frac{dy_j(t)}{dt} = -b_j y_j(t) + \sum_{i=1}^m v_{ij} g_i(x_i(t - \tau_{ij})) + J_j, \end{cases} \quad (2)$$

($i = 1, 2, \dots, m$, $j = 1, 2, \dots, p$), 其中 σ_{ji} 、 τ_{ij} 是非负常数, 表示时滞。这个时滞微分方程组 (2) 就反映了在域 1 (2) 中的神经元 i (j) 在时刻 $t - \tau_{ij}$ ($t - \sigma_{ji}$) 所产生的信号 $g_i(x_i(t - \tau_{ij}))$ ($f_j(y_j(t - \sigma_{ji}))$) 要经过时间 τ_{ij} (σ_{ji}) 后才被域 2 (1) 中的神经元 j (i) 在时刻 t 时被处理。

(3) BAM 神经网络变时滞微分方程模型

Gopalsamy 和 He 研究的一类变时滞 BAM 神经网络形式如下^[3]:

$$\begin{cases} \frac{dx_i(t)}{dt} = -a_i x_i(t) + \sum_{j=1}^p w_{ji} f_j(y_j(t - \sigma_{ji}(t))) + I_i, \\ \frac{dy_j(t)}{dt} = -b_j y_j(t) + \sum_{i=1}^m v_{ij} g_i(x_i(t - \tau_{ij}(t))) + J_j, \end{cases} \quad (3)$$

($i = 1, 2, \dots, m$, $j = 1, 2, \dots, p$), 其中 $\sigma_{ji}(t)$ 、 $\tau_{ij}(t)$ 是非负函数, 表示变时滞。

(4) BAM 神经网络分布时滞微分方程模型

Gopalsamy 和 He 研究的一类用积分 - 微分方程形式表示的具无穷分布时滞的 BAM 神经网络形式如下^[3]:

$$\begin{cases} \frac{dx_i(t)}{dt} = -a_i x_i(t) + \sum_{j=1}^p w_{ji} f_j \left(\int_0^{+\infty} K_{ji}(s) y_j(t-s) ds \right) + I_i, \\ \frac{dy_j(t)}{dt} = -b_j y_j(t) + \sum_{i=1}^m v_{ij} g_i \left(\int_0^{+\infty} H_{ij}(s) x_i(t-s) ds \right) + J_j, \end{cases} \quad (4)$$

($i = 1, 2, \dots, m, j = 1, 2, \dots, p$) 其中 $K_{ji}(s)$ 表示域 2 中第 j 个神经元到域 1 中第 i 个神经元在时刻 s 时触突信号传递时滞核函数, $K_{ji} : [0, +\infty) \rightarrow [0, +\infty)$, $H_{ij}(s)$ 同样表示时滞核函数, $H_{ij} : [0, +\infty) \rightarrow [0, +\infty)$, 它们都是实值非负连续函数, 且满足以下两个条件:

$$\begin{cases} \int_0^{+\infty} K_{ji}(s) ds = 1, \quad \int_0^{+\infty} s K_{ji}(s) ds < \infty, \\ \int_0^{+\infty} H_{ij}(s) ds = 1, \quad \int_0^{+\infty} s H_{ij}(s) ds < \infty. \end{cases} \quad (5)$$

其中 σ_{ji} 、 τ_{ij} 是非负常数, 表示时滞。这个时滞微分方程组就反映了在域 1 (2) 中的神经元 i (j) 在时刻 t 时处理的信号与域 2 (1) 中的神经元 j (i) 在时刻 t 之前所产生的所有信号都有关。这种时滞即是所谓的无穷分布时滞。显然, 对于一个由各种各样体积与长度的轴突所形成的具有众多并行处理通道的神经网络来说, 考虑分布时滞更加合理。

2 时滞微分方程各种数值解算法介绍

2.1 欧拉法算法描述

欧拉法是解初值问题的最简单的数值方法, 对常微分方程初始问题

$$\begin{cases} \frac{dy}{dx} = f(x, y) & (6) \\ y(x_0) = y_0 & (7) \end{cases}$$

用数值方法求解时, 先假定 (6)、(7) 的解存在且唯一。从 (7) 式可知, $y(x_0) = y_0$ 已给定, 因而可以算出 $y'(x_0) = f(x_0, y_0)$, 设 $x_1 = h$ 充分小, 则近似地有:

$$\frac{y(x_1) - y(x_0)}{h} \approx y'(x_0) = f(x_0, y_0) \quad (8)$$

记 $y_i = y(x_i)$ $i = 0, 1, \dots, n$, 从而我们可以取 $y_1 = y_0 + hf(x_0, y_0)$ 作为 $y(x_1)$ 的近似值。

利用 y_1 及 $f(x_1, y_1)$ 又可以算出 $y(x_2)$ 的近似值: $y_2 = y_1 + hf(x_1, y_1)$

一般地, 在任意点 $x_{n+1} = (n+1)h$ 处 $y(x)$ 的近似值由下式给出

$$y_{n+1} = y_n + hf(x_n, y_n) \quad (9)$$

这就是欧拉法的计算公式, h 称为步长。

不难看出，近似解的误差首先是由差商近似代替微商（见(8)）引起的，这种近似代替所产生的误差称为截断误差。还有一种误差称为舍入误差，这种误差是由于利用(8)进行计算时数值舍入引起的。

2.2 改进欧拉法算法描述

为了构造更精确的数值方法，我们从另一角度重新分析一下初值问题。一般说来，一阶方程的初值问题与积分方程

$$y(x) = y_0 + \int_{x_0}^x f(t, y(t)) dt \quad (10)$$

是等价的，当 $x = x_1$ 时，

$$y(x_1) = y_0 + \int_{x_0}^{x_1} f(t, y(t)) dt \quad (11)$$

要得到 $y(x_1)$ 的值，就必须计算出(11)式右端的积分。但积分式中含有未知函数，无法直接计算，只好借助于数值积分。假如用矩形法进行数值积分则

$$\int_{x_0}^{x_1} f(t, y(t)) dt \approx f(x_0, y(x_0))(x_1 - x_0)$$

因此有

$$\begin{aligned} y(x_1) &\approx y_0 + f(x_0, y(x_0))(x_1 - x_0) \\ &= y_0 + hf(x_0, y_0) \end{aligned}$$

可见，用矩形法计算右端的积分与用欧拉法计出的结果完全相同。因此也可以说欧拉法的精度之所以很低是由于采用矩形法计算右端积分的结果。可以想象，用梯形公式计算(11)式右端的积分，可期望得到较高的精度。这时

$$\int_{x_0}^{x_1} f(t, y(t)) dt \approx \frac{1}{2} \{f(x_0, y(x_0)) + f(x_1, y(x_1))\} (x_1 - x_0)$$

将这个结果代入(3)并将其中的 $y(x_1)$ 用 y_1 近似代替，则得

$$y_1 = y_0 + \frac{1}{2} h [f(x_0, y_0) + f(x_1, y_1)]$$

这里得到了一个含有 y_1 的方程式，如果能从中解出 y_1 ，用它作为 $y(x_1)$ 的近似值，可以认为比用欧拉法得出的结果要好些。仿照求 y_1 的方法，可以逐个地求出 y_2, y_3, \dots 。一般地当求出 y_n 以后，要求 y_{n+1} ，则可归结为解方程：

$$y_{n+1} = y_n + \frac{h}{2} [f(x_n, y_n) + f(x_{n+1}, y_{n+1})]$$

这个方法称为梯形法则。用梯形法则求解，需要解含有 y_{n+1} 的方程式，这常常很不容易。为此，在实际计算时，可将欧拉法与梯形法则相结合，计算公式为

$$\begin{cases} y_{n+1}^{(0)} = y_n + hf(x_n, y_n) \\ y_{n+1}^{(k+1)} = y_n + \frac{h}{2} [f(x_n, y_n) + f(x_{n+1}, y_{n+1}^{(k)})] \end{cases} \quad k = 0, 1, 2, \dots \quad (12)$$

这就是先用欧拉法由 (x_n, y_n) 得出 $y(x_{n+1})$ 的初始近似值 $y_{n+1}^{(0)}$ ，然后用 (12) 中第二式进行迭代，反复改进这个近似值，直到 $|y_{n+1}^{(k+1)} - y_{n+1}^{(k)}| < \varepsilon$ (ε 为所允许的误差) 为止，并把 $y_{n+1}^{(k)}$ 取作 $y(x_{n+1})$ 的近似值 y_{n+1} 。这个方法就叫改进欧拉方法。

显然，应用改进欧拉法，如果序列 $y_{n+1}^{(0)}, y_{n+1}^{(1)}, \dots$ 收敛，它的极限便满足方程

$$y_{n+1} = y_n + \frac{h}{2} [f(x_n, y_n) + f(x_{n+1}, y_{n+1})]$$

即序列的极限可取作 y_{n+1} 。可以证明，如果 dy/dx 有界，则只要 h 取得适当小，上述序列必定收敛。这样当 h 取得充分小，就可保证上述迭代过程收敛到一个解。当步长 h 取得适当时，欧拉方法算出的值已是较好的近似，因此改进欧拉法收敛很快，通常只需二、三次迭代即可。如果迭代很多步仍不收敛，这表明步长 h 选的过大，应缩小步长后再计算。通常把 (12) 叫做预报校正公式，其中第一式叫预报公式，第二式叫校正公式。这个公式还可写为

$$\begin{cases} y_{n+1} = y_n + \frac{1}{2}k_1 + \frac{1}{2}k_2 \\ k_1 = hf(x_n, y_n) \\ k_2 = hf(x_n + h, y_n + k_1) \end{cases} \quad (12)'$$

2.3 龙格—库塔法算法描述

由上节知道，截断误差的阶是衡量一个方法精度高低的主要依据。能否用提高截断误差阶来提高方法的精确度呢？回答是肯定的。本节介绍的龙格—库塔法就是基于这种思想构造出来的。

从理论上讲，只要函数 $y = y(x)$ 在区间 $[a, b]$ 上充分光滑，那么它的各阶导数值 $y^{(k)}(x_n)$ 与函数 $y(x)$ 在区间 $[a, b]$ 上某些点的值就相互有联系，就是说，函数值可用各阶导数值近似地表示出来，反之，各阶导数值也可用函数在一些点上值的线性组合近似地表示出来。事实上，欧拉法和改进欧拉法也可以看成是导数值用函数值的线性组合表示的特例，例如，改进欧拉法可以写成 (12)'，此公式也可称为二阶龙格—库塔公式。

为了导出龙格—库塔法的一般公式，我们取如下的线性组合形式，

$$y_{n+1} = y_n + h \sum_{i=1}^r w_i k_i \quad (13)$$

其中

$$k_i = f\left(x_n + b_i h, y_n + h \sum_{j=1}^{i-1} a_{ij} k_j\right) \quad (14), \text{ 即}$$

$$\begin{cases} k_1 = f(x_n, y_n) \\ k_2 = f(x_n + b_2 h, y_n + h a_{21} k_1) \\ k_3 = f(x_n + b_3 h, y_n + h a_{31} k_1 + h a_{32} k_2) \\ \dots\dots \end{cases} \quad (15)$$

而 $w_1, w_2, \dots, w_v; b_1=0, b_2, b_3, \dots, b_v; a_{21}, a_{29}, \dots, a_{v-1}$ 除 $b_1=0$ 外均为待定系数。适当选取这些系数，使得局部截断误差的阶尽可能高即可。

显然，当 $\gamma = 1$ 时，(13) 式就是欧拉公式。

下面我们先导出 $\gamma = 2$ 时的公式。将 k_1, k_2 在同一点 (x_n, y_n) 泰勒展开，则有

$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_2 &= f(x_n + b_2 h, y_n + h a_{21} k_1) \\ &= f(x_n, y_n) + h \left(b_2 \cdot \frac{\partial f}{\partial x} \Big|_{(x_n, y_n)} + a_{21} k_1 \cdot \frac{\partial f}{\partial y} \Big|_{(x_n, y_n)} \right) + O(h^2) \end{aligned} \quad (16)$$

将(16)代入(13)并与 $y(x_n+h)$ 在 x_n 点的泰勒展开式：

$$\begin{aligned} y(x_n + h) &= y(x_n) + h f(x_n, y_n) + \frac{h^2}{2} \left(\frac{\partial f}{\partial x} \Big|_{(x_n, y_n)} + f(x_n, y_n) \cdot \frac{\partial f}{\partial y} \Big|_{(x_n, y_n)} \right) \\ &\quad + \frac{h^3}{3!} \left[\frac{\partial^2 f}{\partial x^2} \Big|_{(x_n, y_n)} + 2 f(x_n, y_n) \frac{\partial^2 f}{\partial x \partial y} \Big|_{(x_n, y_n)} + f^2(x_n, y_n) \frac{\partial^2 f}{\partial y^2} \Big|_{(x_n, y_n)} \right. \\ &\quad \left. + \frac{\partial f}{\partial y} \Big|_{(x_n, y_n)} \left(\frac{\partial f}{\partial x} \Big|_{(x_n, y_n)} + f(x_n, y_n) \frac{\partial f}{\partial y} \Big|_{(x_n, y_n)} \right) \right] + \dots \end{aligned}$$

逐项比较，令 h, h^2 项的系数相等，便得到

$$w_1 + w_2 = 1 \quad w_2 b_2 = \frac{1}{2} \quad w_2 a_{21} = \frac{1}{2}$$

把 b_2 作为自由参数来确定 w_1 和 w_2 ，如取 $b_2 = 1$ ，则 $w_1 = w_2 = \frac{1}{2}$ ， $a_{21} = 1$ ，这时 (13)

正好就是改进的欧拉方法，截断误差的阶为 $O(h^3)$ 。对于 $\gamma = 3$ 的情形，我们也可以完全仿上述方法推导出三阶龙格—库塔公式。这时参数满足下列条件

$$\begin{cases} w_1 + w_2 + w_3 = 1 \\ b_2 w_2 + b_3 w_3 = \frac{1}{2} \\ a_{21} w_2 (a_{31} + a_{32}) w_3 = \frac{1}{2} \\ b_2^2 w_2 + b_3^2 w_3 = \frac{1}{3} \end{cases} \quad (17)$$

$$\begin{cases} b_2 a_{21} w_2 + b_2 (a_{31} + a_{32}) w_3 = \frac{1}{3} \\ a_{21}^2 w_2 + (a_{31} + a_{32})^2 w_3 = \frac{1}{3} \\ b_2 a_{32} w_2 = \frac{1}{6} \\ a_{21} a_{32} w_3 = \frac{1}{6} \end{cases}$$

(17) 比较简单的一组解为:

$$b_2 = \frac{1}{2}, \quad b_3 = 1, \quad a_{21} = \frac{1}{2}, \quad a_{29} = -1, \quad a_{30} = 2, \quad w_1 = \frac{1}{6}, \quad w_2 = \frac{4}{6}, \quad w_3 = \frac{1}{6}$$

将它代入 (13) 得

$$\begin{cases} y_{n+1} = y_n + \frac{h}{6}(k_1 + 4k_2 + k_3) \\ k_1 = f(x_n, y_n) \\ k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right) \\ k_3 = f(x_n + h, y_n - k_1 + 2k_2) \end{cases} \quad (18)$$

这就是三阶龙格-库塔公式。当然, 参数的不同选取公式 (18) 就有不一样的形式, 但它们的截断误差阶都是 $O(h^4)$ 。

通常人们所说的龙格-库塔法是指四阶而言的。我们可以仿照二阶的情形推导出此公式, 不过太繁杂, 此处从略, 常用的四阶公式是

$$\begin{cases} y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 = f(x_n, y_n) \\ k_2 = f\left(x_n + \frac{1}{2}h, y_n + \frac{h}{2}k_1\right) \\ k_3 = f\left(x_n + \frac{1}{2}h, y_n + \frac{h}{2}k_2\right) \\ k_4 = f(x_n + h, y_n + hk_3) \end{cases} \quad (19)$$

公式 (19) 的截断误差阶为 $O(h^5)$ 。

2.4 半离散法算法描述

在这一节, 我们利用半离散化技术来建立连续型神经网络的离散形式。根据第 1.1.2 引入的具常数时滞、变时滞和分布时滞的 BAM 神经网络模型, 下面分别建立各自的离散形式模型:

2.4.1 变时滞 BAM 神经网络的离散化:

先考虑变时滞 BAM 神经网络 (3) 的离散化: 设 h 是一个正的实数, 表示一致离散步长, $[u]$ 表示实数 u 的整数部分。当时间变量 $t \in [nh, (n+1)h)$, $s \in [lh, (l+1)h)$ ($n \in \mathbf{Z}_0^+$,

$l \in \mathbf{Z}^+$) 时, 记 $p(k) = \begin{cases} m, & k=1 \\ p, & k=2 \end{cases}$, 我们用 $\left[\frac{t}{h}\right]h$ 代替系统(3)中的变量 t , 则 BAM 模型

(3)化为如下近似形式:

$$\frac{dx_{ki}(t)}{dt} = -a_{ki}(nh)x_{ki}(t) + \sum_{j=1}^{p(3-k)} w_{(3-k)j} \left(\left[\frac{t}{h}\right]h \right) f_{(3-k)j} \left(x_{(3-k)j} \left(\left[\frac{t}{h}\right]h - \left[\frac{s}{h}\right]h \right) \right) + I_{ki} \left(\left[\frac{t}{h}\right]h \right) \quad (20)$$

考虑到 $\left[\frac{t}{h}\right] = n$, $\left[\frac{s}{h}\right] = l$, 重写上式为

$$\frac{dx_{ki}(t)}{dt} + a_{ki}(nh)x_{ki}(t) = \sum_{j=1}^{p(3-k)} w_{(3-k)j}(nh) f_{(3-k)j}(x_{(3-k)j}(nh-lh)) + I_{ki}(nh),$$

两边乘以 $e^{a_{ki}(nh)t}$, 得

$$\left(x_{ki}(t) e^{a_{ki}(nh)t} \right)' = \left(\sum_{j=1}^{p(3-k)} w_{(3-k)j}(nh) f_{(3-k)j}(x_{(3-k)j}(nh-lh)) + I_{ki}(nh) \right) e^{a_{ki}(nh)t},$$

在区间 $[nh, t)$ (其中 $t < (n+1)h$) 上对上式积分得

$$\begin{aligned} x_{ki}(t) e^{a_{ki}(nh)t} - x_{ki}(nh) e^{a_{ki}(nh)nh} \\ = \left(\sum_{j=1}^{p(3-k)} w_{(3-k)j}(nh) f_{(3-k)j}(x_{(3-k)j}(nh-lh)) + I_{ki}(nh) \right) \frac{e^{a_{ki}(nh)t} - e^{a_{ki}(nh)nh}}{a_{ki}(nh)}, \end{aligned}$$

令 $t \rightarrow (n+1)h$, 我们得到

$$\begin{aligned} x_{ki}((n+1)h) &= x_{ki}(nh) e^{-a_{ki}(nh)h} \\ &+ \left(\sum_{j=1}^{p(3-k)} w_{(3-k)j}(nh) f_{(3-k)j}(x_{(3-k)j}(nh-lh)) + I_{ki}(nh) \right) \frac{1 - e^{-a_{ki}(nh)h}}{a_{ki}(nh)}. \end{aligned}$$

为方便起见, 我们采用以下记号:

$$x_{ki}(n) = x_{ki}(nh), \quad \alpha_{ki}(n) = e^{-a_{ki}(nh)h}, \quad (21)$$

$$w_{(3-k)j}(n) = w_{(3-k)j}(nh) \frac{1 - e^{-a_{ki}(nh)h}}{a_{ki}(nh)}, \quad I_{ki}(n) = I_{ki}(nh) \frac{1 - e^{-a_{ki}(nh)h}}{a_{ki}(nh)}, \quad (22)$$

则得到变时滞变系数的双向联想记忆离散神经网络:

$$x_{ki}(n+1) = \alpha_{ki}(n)x_{ki}(n) + \sum_{j=1}^{p(3-k)} w_{(3-k)j}(nh) f_{(3-k)j}(x_{(3-k)j}(nh-lh)) + I_{ki}(nh) \quad (23)$$

($n \in \mathbf{Z}_0^+$, $k=1, 2$, $i=1, 2, \dots, p_k$)。显然, 当 $h > 0$ 、 $a_{ki}(t) > 0$ 时, $0 < \alpha_{ki}(n) < 1$ 。可

以证明^[1]，当 $h \rightarrow 0^+$ 时，离散 BAM 神经网络(23)收敛为连续 BAM 神经网络(3)。

离散 BAM 神经网络(23)的初始条件为

$$x_{ki}(l) = \varphi_{ki}(l), \quad (24)$$

其中 $k=1,2$ ， $i=1,2,\dots,p_k$ ，其中 $\varphi_{ki}(\cdot)$ 定义在式(10)中。

对于常数时滞 BAM 神经网络(2)的离散化与变时滞情形类似，唯一的区别是 τ 与 n 无关，所以常数时滞变系数的双向联想记忆离散神经网络模型为

$$x_{ki}(n+1) = \alpha_{ki}(n)x_{ki}(n) + \sum_{j=1}^{p(3-k)} w_{(3-k)j}(nh) f_{(3-k)j}(x_{(3-k)j}(nh-\tau)) + I_{ki}(n) \quad (25)$$

($n \in \mathbf{Z}_0^+$ ， $k=1,2$ ， $i=1,2,\dots,p_k$)。其初始条件与(24)相同。

2.4.2 分布时滞 BAM 神经网络的离散化

这小节讨论连续分布时滞 BAM 神经网络(4)的离散化。当 $t \in [nh, (n+1)h)$ ， $s \in [lh, (l+1)h)$ ($n \in \mathbf{Z}_0^+$ ， $l \in \mathbf{Z}^+$) 时，将系统(4)中的积分项用如下无穷和来近似表示，

$$\begin{aligned} \int_0^{+\infty} g_{(3-k)j}(s)x_{(3-k)j}(t-s)ds &\approx \sum_{\left[\frac{s}{h}\right]=1}^{+\infty} g_{(3-k)j}\left(\left[\frac{s}{h}\right]h\right)x_{(3-k)j}\left(\left[\frac{t}{h}\right]h - \left[\frac{s}{h}\right]h\right)\varphi(h) \\ &= \sum_{l=1}^{+\infty} g_{(3-k)j}(lh)x_{(3-k)j}(nh-lh)\varphi(h), \end{aligned}$$

其中 $\varphi(h) > 0$ (当 $h > 0$)，且 $\varphi(h) \approx h + O(h^2)$ (当 $h > 0$ 较小时)。其它步骤与变时滞离散化情形相同，我们可以得到分布时滞变系数的双向联想记忆离散神经网络模型为

$$x_{ki}(n+1) = \alpha_{ki}(n)x_{ki}(n) + \sum_{j=1}^{p(3-k)} w_{(3-k)j}(n) f_{(3-k)j}\left(\sum_{l=1}^{+\infty} g_{(3-k)j}(l)x_{(3-k)j}(n-l)\right) + I_{ki}(n), \quad (26)$$

($n \in \mathbf{Z}_0^+$ ， $k=1,2$ ， $i=1,2,\dots,p_k$)。其中非负序列

$$g_{(3-k)j}(l) = g_{(3-k)j}(lh)\varphi(h) \quad (l \in \mathbf{Z}^+) \quad (27)$$

应该恰当选取函数 $\varphi(h)$ 使得非负序列 $g_{(3-k)j}(l)$ 满足以下条件：

$$\sum_{l=1}^{+\infty} g_{(3-k)j}(l) = 1, \sum_{l=1}^{+\infty} l g_{(3-k)j}(l) < +\infty.$$

离散 BAM 神经网络(26)的初始条件为

$$x_{ki}(l) = \varphi_{ki}(l), \quad l \in \mathbf{Z}_0^+, \quad (28)$$

其中 $k=1,2$ ， $i=1,2,\dots,p_k$ ，其中 $\varphi_{ki}:(-\infty, 0] \rightarrow \mathbf{R}$ 是连续的。

如果连续型 BAM 神经网络的外部输入、联接权重和信号衰减系数是常数，则其对应的离散模型中 $\alpha_{ki}(n)$ 、 $w_{(3-k)j}(n)$ 、 $I_{ki}(n)$ 都与 n 无关，即式(21)、(22)分别化为

$$\alpha_{ki} = e^{-a_{ki}h}, w_{(3-k)ji} = w_{(3-k)ji}(h) \frac{1 - e^{-a_{ki}h}}{a_{ki}}, I_{ki} = I_{ki}(h) \frac{1 - e^{-a_{ki}h}}{a_{ki}}, \quad (29)$$

相应地就得到了常系数离散 BAM 神经网络模型:

$$x_{ki}(n+1) = \alpha_{ki}x_{ki}(n) + \sum_{j=1}^{n_p} w_{(3-k)j} f_{(3-k)j}(x_{(3-k)j}(n-l)) + I_{ki}, \quad (30)$$

$$x_{ki}(n+1) = \alpha_{ki}x_{ki}(n) + \sum_{j=1}^{p(3-k)} w_{(3-k)j} f_{(3-k)j} \left(\sum_{l=1}^{+\infty} g_{(3-k)j}(l) x_{(3-k)j}(n-l) \right) + I_{ki}. \quad (31)$$

3 时滞微分方程多种数值解法的计算机实现及其可视化处理

3.1 欧拉法、改进欧拉法实例与计算机仿真^[17]

在这里, 我们考虑 BAM 神经网络模型的无时滞微分方程模型 (1) 的仿真过程。其实, 对于无时滞微分方程模型的情况, 可以归纳为一般微分方程的初值问题的解, 下面我们选用了—个比较典型的例子来进行分析, 解答和仿真, 并通过仿真图来反映欧拉法、改进欧拉法算法的性能特点和差异。

例题 1 用 Euler 方法和改进 Euler 方法解微分方程

$$\begin{cases} \frac{dy}{dx} = -y + x + 1 \\ y(x_0) = 1, x_0 = 0 \end{cases} \quad (32)$$

$$y(x_0) = 1, x_0 = 0 \quad (33)$$

解 取 $h = 0.1$, Euler 方法为

$$y_{k+1} = \frac{9}{10}y_k + \frac{k}{100} + \frac{1}{10} \quad \text{改进的 Euler 方法为}$$

$$y_{k+1} = 0.905y_k + 0.0095k + 0.1$$

计算结果与准确解 $y(x) = e^{-x} + x$ 比较, 列在表表 1 中:

表 1 例 1 的计算结果

Table 1 The computational result of e.g. 1

x_k	y_k	Euler 方法		改进欧拉法	
		y_k	$ y_k - y(x_k) $	y_k	$ y_k - y(x_k) $
0.0	1.000000	1.000000	0.0	1.000000	0.0
0.1	1.004837	1.000000	4.8×10^{-3}	1.005000	1.6×10^{-4}
0.2	1.018730	1.010000	8.7×10^{-3}	1.019025	2.9×10^{-4}
0.3	1.040818	1.029000	1.2×10^{-2}	1.041218	4.0×10^{-4}

0.4	1.070320	1.056100	1.4×10^{-2}	1.070800	4.8×10^{-4}
0.5	1.106530	1.090490	1.6×10^{-2}	1.107076	5.5×10^{-4}
0.6	1.148811	1.131441	1.7×10^{-2}	1.149404	5.9×10^{-4}
0.7	1.196585	1.178297	1.8×10^{-2}	1.197210	6.2×10^{-4}
0.8	1.249328	1.230467	1.9×10^{-2}	1.249975	6.5×10^{-4}
0.9	1.306569	1.287420	1.9×10^{-2}	1.307228	6.6×10^{-4}
1.0	1.367879	1.348678	1.9×10^{-2}	1.368514	6.6×10^{-4}

从表中可以看出，随着 n 的增大，误差也在增大，所以说，欧拉法计算简便，对一些问题有较大的使用价值，但是，它的误差较大，所得的数值解精确度不高，对上述解分别进行仿真，结果如图所示：

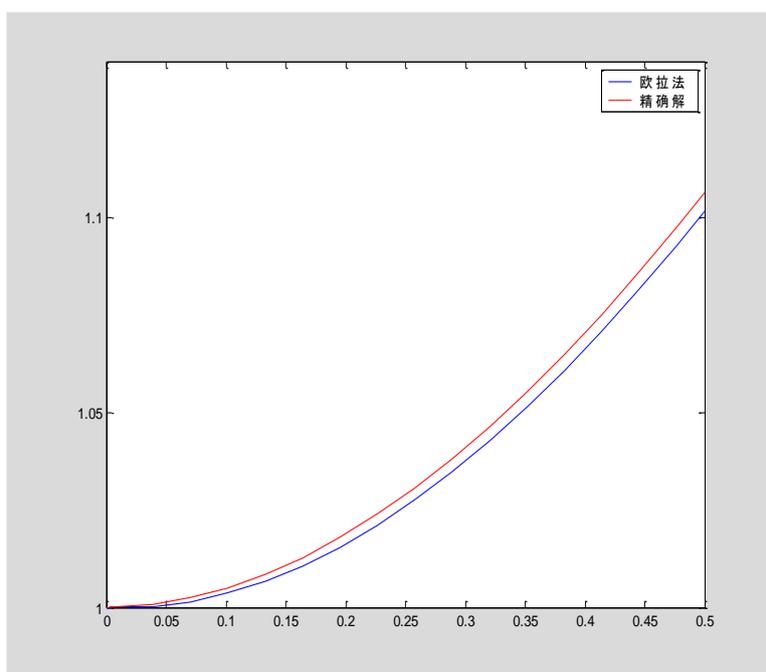


图 1 欧拉离散数值解法仿真图

Fig.1 The map of simulation with Euler method

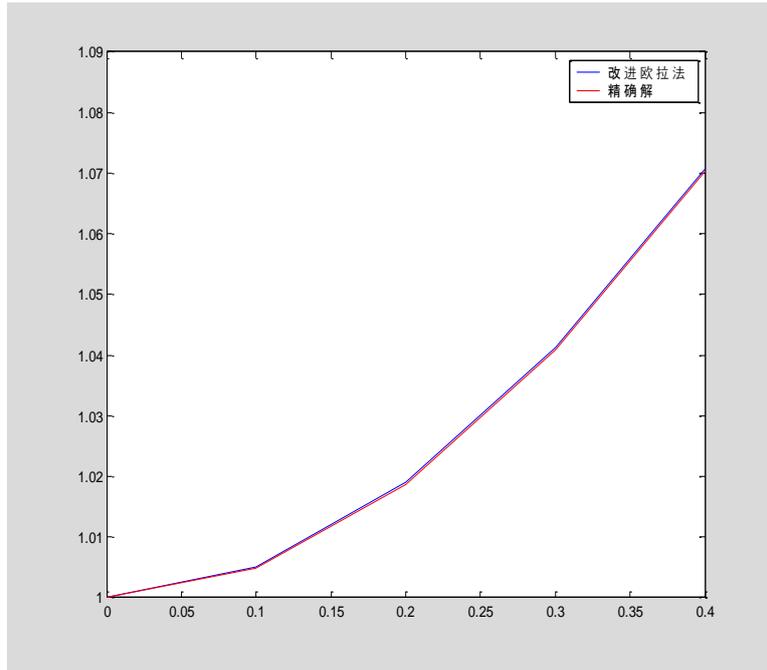


图 2 改进欧拉离散数值解法仿真图

Fig.2 The map of simulation with improved Euler method

3.1 龙格-库塔法算法实例与计算机仿真^[18]

在这里，我们考虑 BAM 神经网络模型的常数时滞微分方程模型 (2) 的仿真过程，通过仿真图来反映龙格-库塔法算法的性能特点^[4]：

例题 2

$$\begin{cases} \frac{dx(t)}{dt} = -6x(t) - 5/2(y(t-\tau)) + 20 & (34) \\ \frac{dy(t)}{dt} = -6y(t) + 3/2(x(t-\tau)) + 30 & (35) \end{cases}$$

这里 $\tau = 1.1499$ ， $x(0) = 0.5, y(0) = 0.5$ ，根据显示的二阶和三阶龙格-库塔法，利用 MatLab 自带的 DDE 算法，对此时滞微分方程进行仿真，结果如图 3 所示：

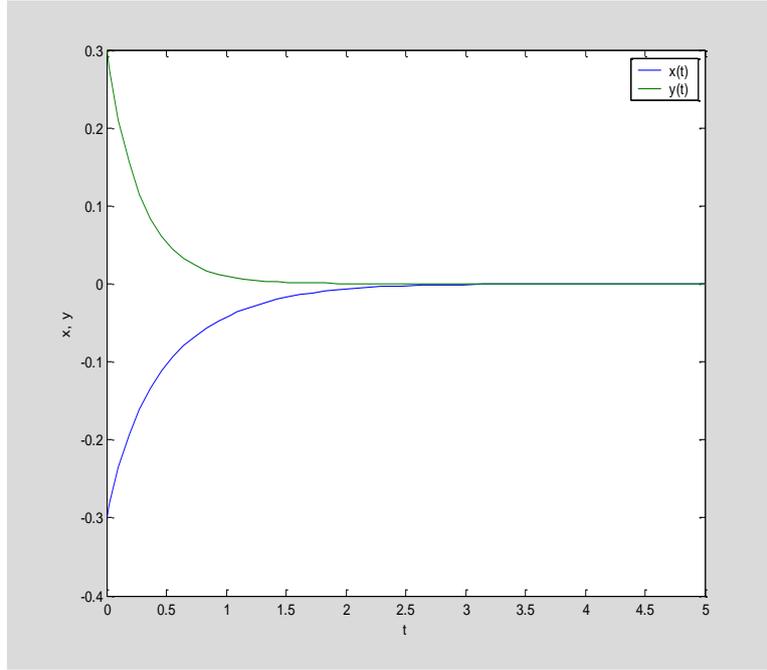


图 3 龙格-库塔法算法离散数值解法仿真图

Fig.3 The map of simulation with Runge - Kutta method

3.3 半离散法算法实例与计算机仿真

在这里，我们考虑 BAM 神经网络模型的分布时滞微分方程模型（4）的仿真过程，通过仿真图来反映分布时滞微分方程模型（4）经过半离散算法离散化后的方程解的性能特点。

例 3 考虑 BAM 神经网络模型（4）通过第 2 节的半离散化方法离散化后的模型：

$$\begin{cases} x_{11}(n+1) = \alpha_{11}(n)x_{11}(n) + w_{21}(n)f_{21}\left(\sum_{s=1}^{+\infty} g_{21}(s)x_{21}(n-s)\right) + I_{11}(n), \\ x_{21}(n+1) = \alpha_{21}(n)x_{21}(n) + w_{11}(n)f_{11}\left(\sum_{s=1}^{+\infty} g_{11}(s)x_{11}(n-s)\right) + I_{21}(n), \end{cases} \quad (36)$$

其中 $\alpha_{11}(n) = \frac{11}{12}$, $\alpha_{21}(n) = \frac{25}{12}$, $I_{11}(n) = \sin b\pi n$, $I_{21}(n) = \cos b\pi n$, $w_{11}(n) = 3$, $w_{21}(n) = \frac{1}{3}$, $g_{11}(n) = g_{21}(n) = (\exp(2)-1)/2 * \exp(-2*n)$, 在神经网络（36）中取无界函数 $f_{ki}(x) = x$, $k=1,2$, 通过相关定理可以知道^{[1][7][8][10]}, BAM 神经网络（36）存在一个全局指数稳定的周期解,数值仿真结果如图 4。

数值仿真时，取 $b=8$ ，并取三种初始状态如下：

$$\begin{aligned} x_{11}(n) &= 0.6 + 0.2 \cos(2nb\pi), x_{21}(n) = 0.6 + 0.4 \sin(3nb\pi), \\ y_{11}(n) &= -0.2 + 0.8 \sin(3nb\pi), y_{21}(n) = 0.6 + 0.5 \cos(nb\pi), (n \in \mathbb{Z}_0^-), \\ z_{11}(n) &= 0.2, z_{21}(n) = -0.2 \end{aligned} \quad (37)$$

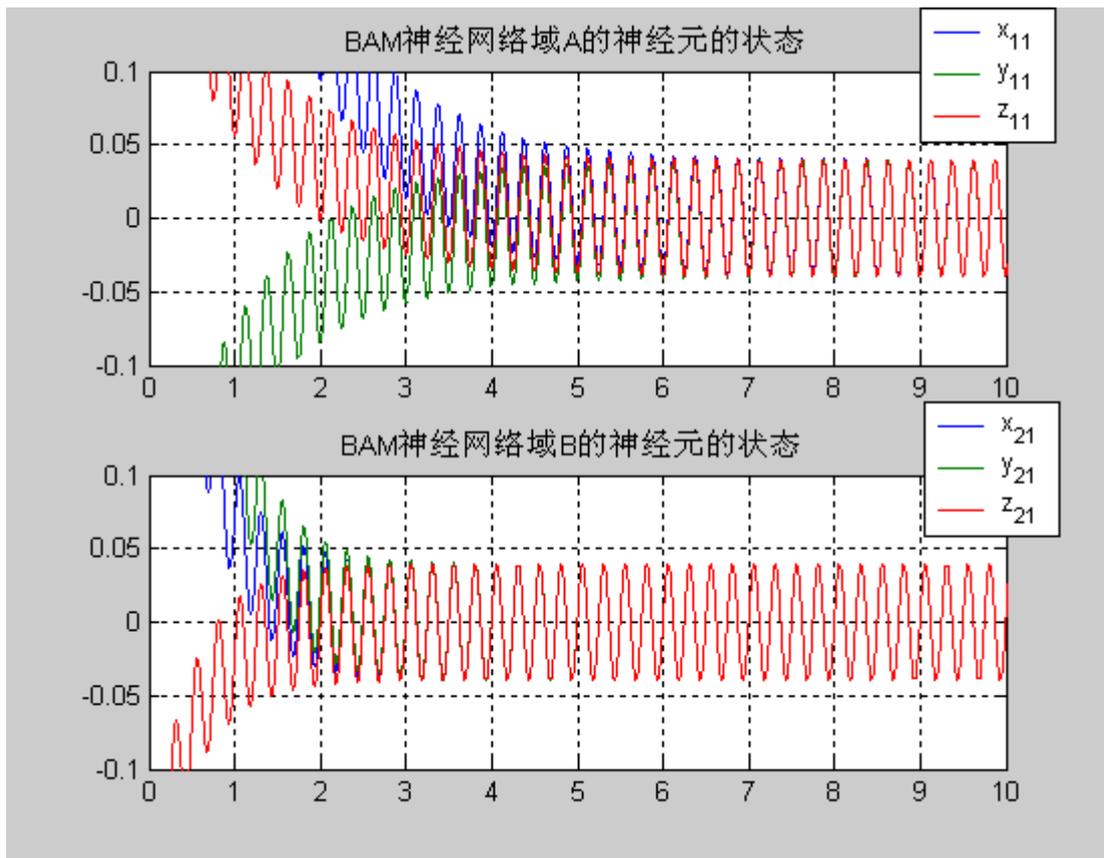


图 4 离散神经网络 (36) 在三种初始条件 (37) 下周期解的全局指数稳定性

Fig 4 The global exponential stability of the periodic solution of the discrete-time neural network (36) under the initial conditions (37)

4 时滞微分方程解各种数值解法性能对比分析

4.1 欧拉法、改进欧拉法和龙格-库塔法性能分析^[19]

首先考虑欧拉法、改进欧拉法两个公式的截断误差： $y(x_{n+1}) - y_{n+1}$ 有多大？这里假定前一步得的结果 $y_n = y(x_n)$ 是准确的，写出 $y(x_{n+1})$ 的泰勒展开式为

$$y(x_{n+1}) = y(x_n + h) = y(x_n) + hy'(x_n) + \frac{h^2}{2!} y''(x_n) + \dots$$

由欧拉法得 $y_{n+1} = y_n + hf(x_n, y_n) = y(x_n) + hy'(x_n)$

$$\text{两式相减得 } y(x_{n+1}) - y_{n+1} = \frac{h^2}{2} y''(x_n) + \dots = O(h^2)$$

即欧拉法的截断误差为 $O(h^2)$ ，当 $h \rightarrow 0$ 时它与 h^2 是同阶无穷小量。

对于改进的欧拉方法，我们以迭代一次的预报校正格式(12)'为例来说明。因为

$$\begin{aligned} k_1 &= hf(x_n, y_n) = hf(x_n, y(x_n)) \\ &= hy'(x_n) \end{aligned}$$

$$\begin{aligned}
k_2 &= hf(x_n + h, y_n + k_1) \\
&= hf[x_n + h, y(x_n) + k_1] \\
&= h \left\{ f[x_n, y(x_n)] + h \frac{\partial}{\partial x} f[x_n, y(x_n)] + k_1 \frac{\partial}{\partial y} f[x_n, y(x_n)] + \dots \right\} \\
&= hf[x_n, y(x_n)] + h^2 \left\{ \frac{\partial}{\partial x} f[x_n, y(x_n)] \right. \\
&\quad \left. + y'(x_n) \frac{\partial}{\partial y} f[x_n, y(x_n)] \right\} + \dots \\
&= hy'(x_n) + h^2 y''(x_n) + \dots
\end{aligned}$$

用它们代入 (12) 式第二式即得

$$\begin{aligned}
y_{n+1} &= y_n + \frac{h}{2} y'(x_n) + \frac{h}{2} y'(x_n) + \frac{h^2}{2} y''(x_n) + \dots \\
&= y(x_n) + hy'(x_n) + \frac{h^2}{2} y''(x_n) + \dots
\end{aligned}$$

这里未把含有 h 的三次幂以上的项写出, 因此有 $y(x_{n+1}) - y_{n+1} = O(h^3)$, 即迭代一次的预报校正格式 (12) 的截断误差为 $O(h^3)$ 。可见改进的欧拉方法比欧拉法的阶提高了。

其次, 通过比较, 可以发现 Runge-Kutta 方法的结果比 Euler 方法、改进的 Euler 方法好得多。在相同步长下, Runge-Kutta 方法的计算量是 Euler 方法的 4 倍, 改进的 Euler 方法的二倍。若 Runge-Kutta 方法步长为 h , Euler 方法步长取 $h/4$, 改进的 Euler 方法取 $h/2$, 它们的计算量将大致相等, 但 Runge-Kutta 方法仍比 Euler 方法、改进的 Euler 方法好得多。

从仿真的结果我们可以明显得知以上的结论。

4. 2 半离散算法性能分析

将连续型系统进行离散化, 我们已经讨论可常规的方法欧拉法(Euler)、龙格-库塔法(Runge-Kutta)等数值解法, 但是离散化后的系统的动力学性质可能与它对应的连续系统所拥有的性质十分不同, 这样可能导致我们在研究连续动力系统时所获得的一些结论, 如平衡点的存在性、稳定性的判别准则不能使用。例如, 有关文献中, 我们了解到连续 BAM 神经网络的一个指数稳定的平衡点在经过标准欧拉法离散后不再指数稳定的现象。如果这种现象发生, 则用离散系统来仿真、实验或计算连续系统就失去了实际意义。对于半离散化技术, 已被广泛地应用于连续动力系统的离散化, 并已经从理论上证明了离散化后的网络与其连续系统拥有相同的动力学性质。基于该方法所表现的优点, 本文通过采用这种技术对 BAM 神经网络进行离散化, 通过上面对连续系统进行离散化过程, 我们可以看到离散系统继承了它对应的连续系统的动力学性质。

5 结论

通过以上研究过程,我们运用不同数值解法对其解的性能进行对比研究,并利用 Matlab 在计算机实现这些解法,对各种解的性能进行可视化仿真处理,最后根据仿真结果针对这些时滞微分方程简单地讨论了他们的平衡点、周期解的存在性与稳定性,非常直观地得到了他们解的一些性质差异。此外,本文所采用的对比研究和数值可视化研究方法为我们今后研究问题提供了一个非常好的方法。

参考文献

- [1] Zhou TJ, Liu YR, Liu YH. Existence and global exponential stability of periodic solution for discrete-time BAM neural networks. *Appl Math Comput* 2006; 182 (2): 1341-1354.
- [2] Kosto, Bidirectional associative memories, *IEEE Trans. Syst. Man Cybernet* 18(1)(1988)49-60
- [3] Gopalsamy, X.Z. He, Delay-independent stability in bidirectional associative memory networks, *IEEE Trans. Neural Networks* 5 (6) (1994) 998-1002
- [4] Jinling Liang, Jinde Cao. Exponential stability of continuous-time and discrete-time bidirectional associative memory networks *Chaos, Solitons and Fractals* 22(2004) 773-785
- [5] LAMBERT J D. *Computational methods in ordinary differential equations*[M]. London: John Wiley Sons Ltd, 1973. 116.
- [6] Hale J K, Verduyn L S M. *Introduction to Functional Differential Equations*[M]. New York: Springer-Verlag, 1993
- [7] L Torelli. Stability of numerical methods for delay differential equations. *J. Comput. Appl. Math.* 1989, 25(1) :15~26 .
- [8] Wang Quanyi. Existence, Stability and Uniqueness of Periodic Solutions. *Chinese Annals of Mathematics*. 1994, 15A(5) :537-545 .
- [9] Mohamad S, Naim AG. Discrete-time analogues of integrodifferential equations modeling bidirectional neural networks. *J Comput Appl Math* 2002; 138: 1-20.
- [10] Li X L, Zhu D M. Global Existence of Positive Periodic Solutions for a Distributed Delay Competition Model [J]. *Acta Mathematica Applicatae Sinica, English Series*, 2003, 19 (3) :491-498
- [11] 李庆扬、王能超、易大义.数值分析[M].第四版.北京:清华大学出版社, Springer, 2001
- [12] 王高雄,周之铭,朱思铭等.常微分方程[M].北京:高等教育出版社, 1983
- [13] 胡中波,苏清华.常微分方程数值解法的一个一般公式. *赣南师范学院学报*, 2003(6) .

- [14] 冯元珍, 屠小明. 基于 MATLAB 的微分方程数值解法[J]. 科技信息(学术版), 2006,(12)
- [15] 尚智, 刘瑞兰, 苏光辉, 贾斗南. 微分方程的神经网络数值解法[J]. 计算物理, 2001,(03)
- [16] 马燕, 张根耀, 杜利锋, 李竹林. 常微分方程的数值解法及其在计算机中的实现[J]. 延安大学学报(自然科学版), 2000,(02).
- [17] 张丽娟. 常微分方程的 Euler 解法及其计算机实现[J]. 长春师范学院学报, 2005,(12)
- [18] 林立军, 郭松云. 常微分方程数值解法——Runge—Kutta 法的历史浅析. 辽宁师范大学学报: 自然科学版, 2003(2).
- [19] 徐应祥. 关于常微分方程初值问题数值解误差的探讨[J]. 西北师范大学学报(自然科学版), 2003,(01)
- [20] 王沫然. MATLAB 与科学计算[M]. 北京: 电子工业出版社, 2005.

致 谢

首先，感谢关心和指导我的老师和同学，特别感谢周铁军教授和寝室的同学们！在本次毕业论文设计过程中，通过与老师们的沟通和与同学们的交流，我深刻地感觉到了理论基础知识对指导科研工作的重要性，并发现独立自主、沟通交流与信息摄取的能力对于提高自己的创新能力的至关重要性；其次，感谢工作在理论前线的科技工作者们，正是他们优秀的工作成果，为我打开了广阔的视野，使我可以新的不同的角度来分析和处理课题；最后，感谢大量的无私的网络信息提供者，正是他们的积极工作，使我能够及时猎涉到所需的大量信息，圆满地完成我的毕业论文！总之，感谢所有对我提供诚挚无私的帮助的人们，在此我对他们致以深深的敬意和衷心的感谢！

后记：通过本次论文设计，在具体的知识学习方面，我学到了很多新的知识：通过广泛阅读常微分方程方面的书籍和大量浏览相关网络资料，比较全面地理解了常微分领域数值解法的知识，通过引入神经网络模型，熟悉了很多相关的神经网络的知识，通过用 MatLab 对建立的神经网络模型进行编程实现相应的数值算法和可视频化处理，大大提高了自己的 MatLab 编程实践能力。总之，在整个毕业论文设计过程中我培养了更加严谨、认真的态度，更强的分析问题、解决问题的能力 and 更加积极的处事态度！