

时滞微分方程多种数值解法的 计算机实现

学生姓名：熊君武

指导老师：周铁军 教授

学 院：理学院

- 一 选题背景综述
- 二 双向联想记忆神经网络模型
- 三 时滞微分方程各种数值解算法介绍
- 四 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
- 五 时滞微分方程解各种数值解法性能对比分析
- 六 结论

一 选题背景综述

- 1 研究理论：

常微分方程概念、解法、理论：

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，通解以及通过通解求特解等。

- 2 研究对象：

能求出通解的情况不多，多是求满足某种指定条件的特解。而且只有极少数的类型可以用初等函数显式地写出精确解析解，大多数情况下我们必须寻求解析形式的容许一定误差的数值解，即通过加减乘除和函数运算，求得特定点的近似解。

- 3 研究方法:

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

- 4 研究工具:

利用Mat I Lab在计算机实现多种数值解法并其绘制图形,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

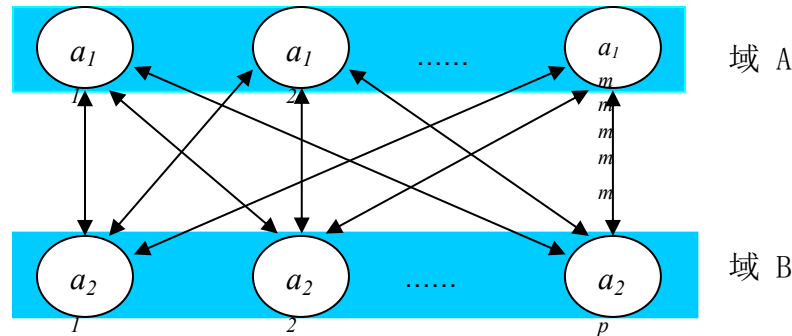
- 5 研究意义:

通过不同方法的数值仿真,对比它们的性能。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。

二 双向联想记忆神经网络模型

- 1 双向联想记忆 (BAM) 神经网络的概念

是神经元分布在两个域上，两域神经元之间进行全互联，但同一域的神经元之间互相没有连接，其拓扑结构如图所示。



图中，BAM神经网络的域A中有 m 个神经元，即 $F1 = \{a11, a12, \dots, a1m\}$ ，域B中有 p 个神经元，即 $F2 = \{a21, a22, \dots, a2p\}$ 。它可以存储成对模式或记忆，通过采用前向和反向式的双向搜索，可从一个输入对 (A, B) 回忆一相关的双极性向量对 (Ak, Bk) ，实现异联想记忆功能。

2 双向联想记忆神经网络模型

(1) BAM神经网络无时滞微分方程模型

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

(2) BAM神经网络常数时滞微分方程模型

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

(3) BAM神经网络变时滞微分方程模型

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

(4) BAM神经网络分布时滞微分方程模型

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

三 时滞微分方程各种数值算法介绍

1 欧拉算法

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

2 改进欧拉法算法

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

3 龙格—库塔法算法

(1) 龙格-库塔 (Runge - Kutta) 法的一般形式

$$\left\{ \begin{array}{l} y_{m+1} = y_m + h \sum_{i=1}^r \omega_i K_i \\ K_i = f(t_m + a_i h, y_m + \sum_{j=1}^{i-1} b_{ij} K_j) \\ i = 1, 2, \dots, r \end{array} \right.$$

(2) 二阶Runge - Kutta法公式

$$\left\{ \begin{array}{l} K_1 = f(t_m, y_m) \\ K_2 = f(t_m + h, y_m + K_1 h) \\ y_{m+1} = y_m + \frac{h}{2} (K_1 + K_2) \end{array} \right.$$

(3) 三阶Runge - Kutta公式

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

4 半离散法算法描述

(1) 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神经网络的连续形式如下：

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

(2) 变时滞微分方程模型半离散化算法描述

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 属于 $(nh, (n+1)h)$ ，

s 属于 $(lh, (l+1)h)$ (n, l 属于正整数) 时，记 $p(k) = \begin{cases} m, & k=1 \\ p, & k=2 \end{cases}$

我们用 $[t/h]h$ 代替系统(1)中的变量 t ，则BAM模型(1)化为如下近似形式：

半离散化后的离散形式如下：

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

$[t/h]=n$, $[s/h]=l$, 重写上式：

$$\frac{dx_{ki}(t)}{dt} + a_{ki}(nh)x_{ki}(t)$$

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

两边乘以 $e^{a_{ki}(nh)t}$

$$\left(x_{ki}(t) e^{a_{ki}(nh)t} \right)' =$$

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

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

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

令 $t \rightarrow (n+1)h$, 得到下式:

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

我们采用以下记号：

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

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

则得到变时滞变系数的双向联想记忆离散神经网络：

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

(3) 常数时滞变系数的双向联想记忆离散神经网络模型

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

(4) 分布时滞变系数的双向联想记忆离散神经网络模型

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)

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

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

四 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

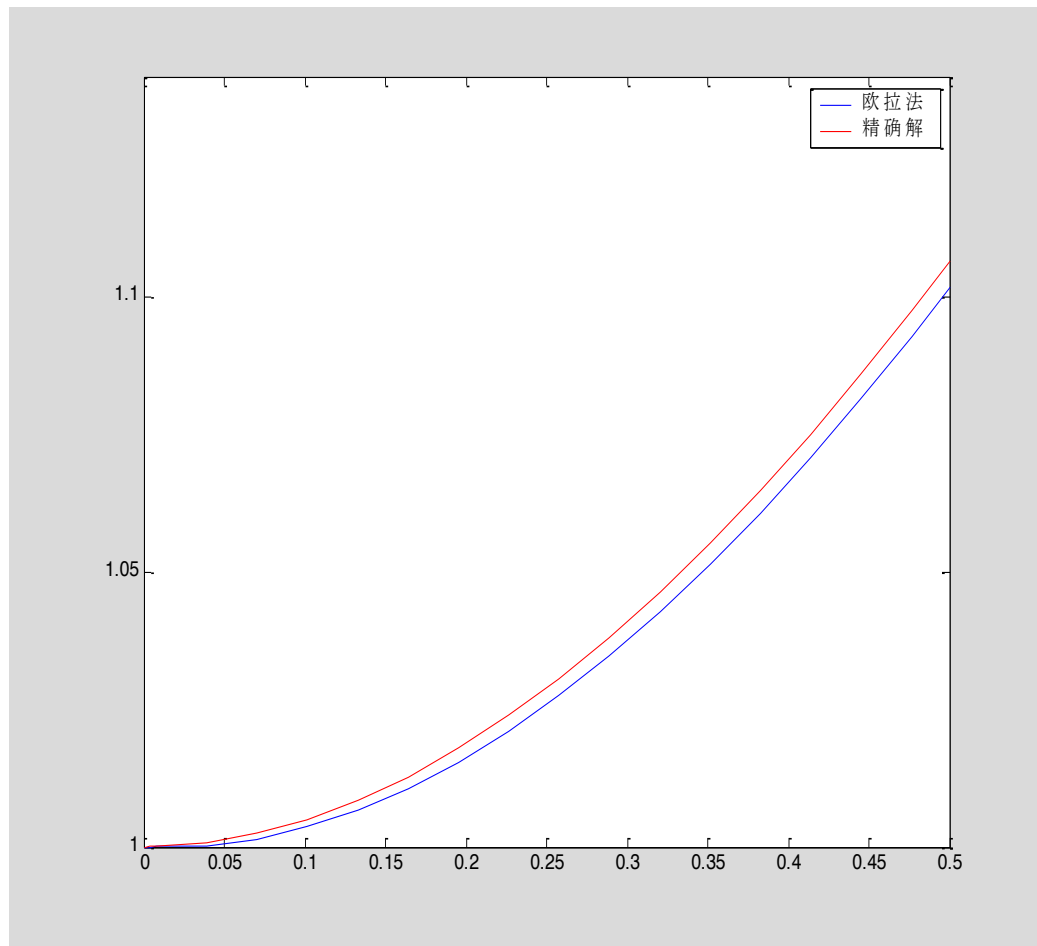
1 欧拉法、改进欧拉法实例与计算机仿真

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

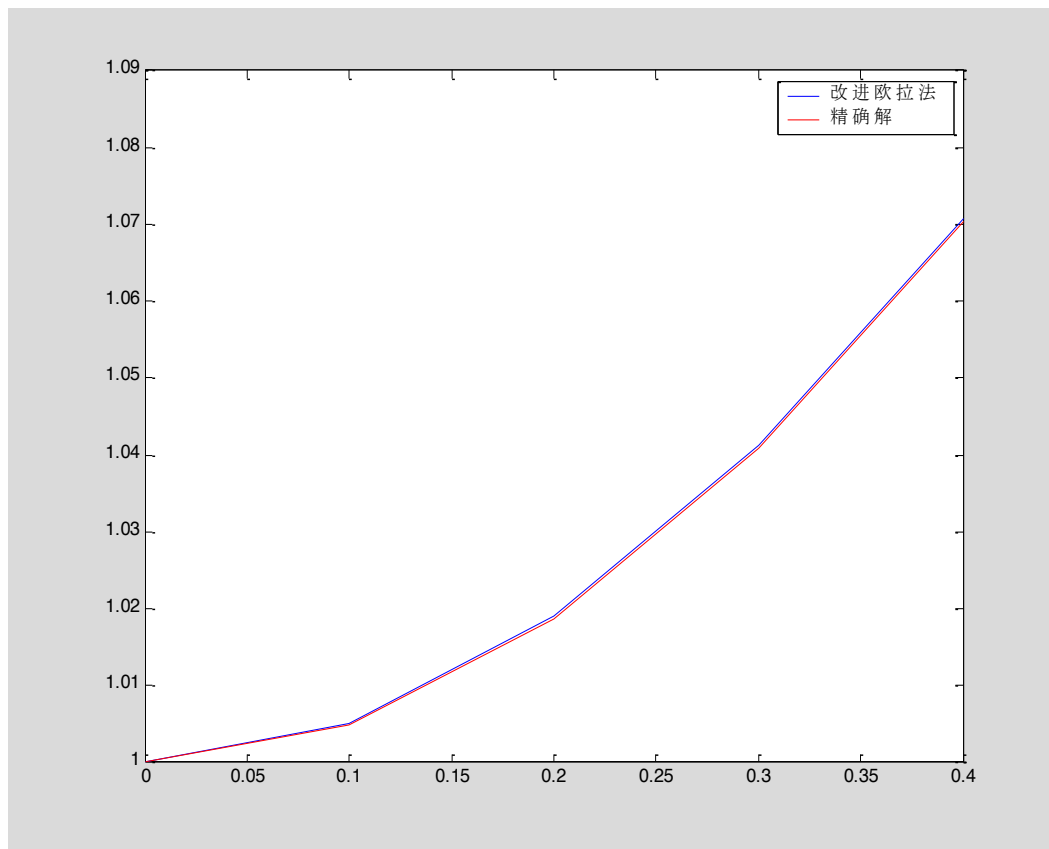
例题1 用Euler方法和改进Euler方法解微分方程

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

(1) 欧拉离散数值解法仿真图



(2) 改进欧拉离散数值解法仿真图



2 龙格—库塔法算法实例与计算机仿真

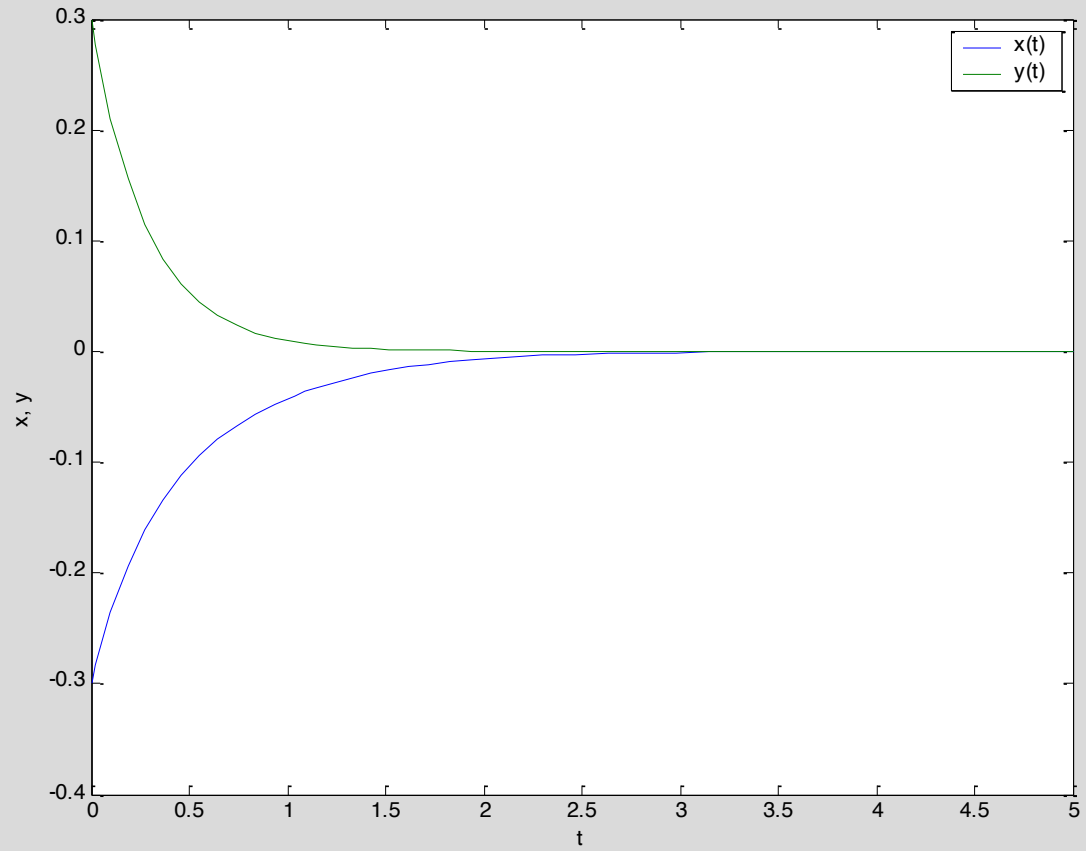
在这里，我们考虑BAM神经网络模型的常数时滞微分方程模型（2）的仿真过程：

例题2

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

这里 $\tau = 1.1499$, $x(0) = 0.5, y(0) = 0.5$

龙格—库塔法算法离散数值解法仿真图



(3) 半离散法算法实例与计算机仿真

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

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：

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

其中 $\alpha_{11}(n) = \frac{11}{12}$, $\alpha_{21}(n) = \frac{25}{12}$, $I_{11}(n) = \sin b\pi n$, $I_{21}(n) = \cos b\pi n$

$w_{11}(n) = 3$, $w_{21}(n) = \frac{1}{3}$, $g_{11}(n) = g_{21}(n) = (\exp(2)-1)/2 * \exp(-2*n)$

传输函数： $f_{ki}(x) = x$, $k=1,2$,

数值仿真时，取，并取三种初始状态如下：

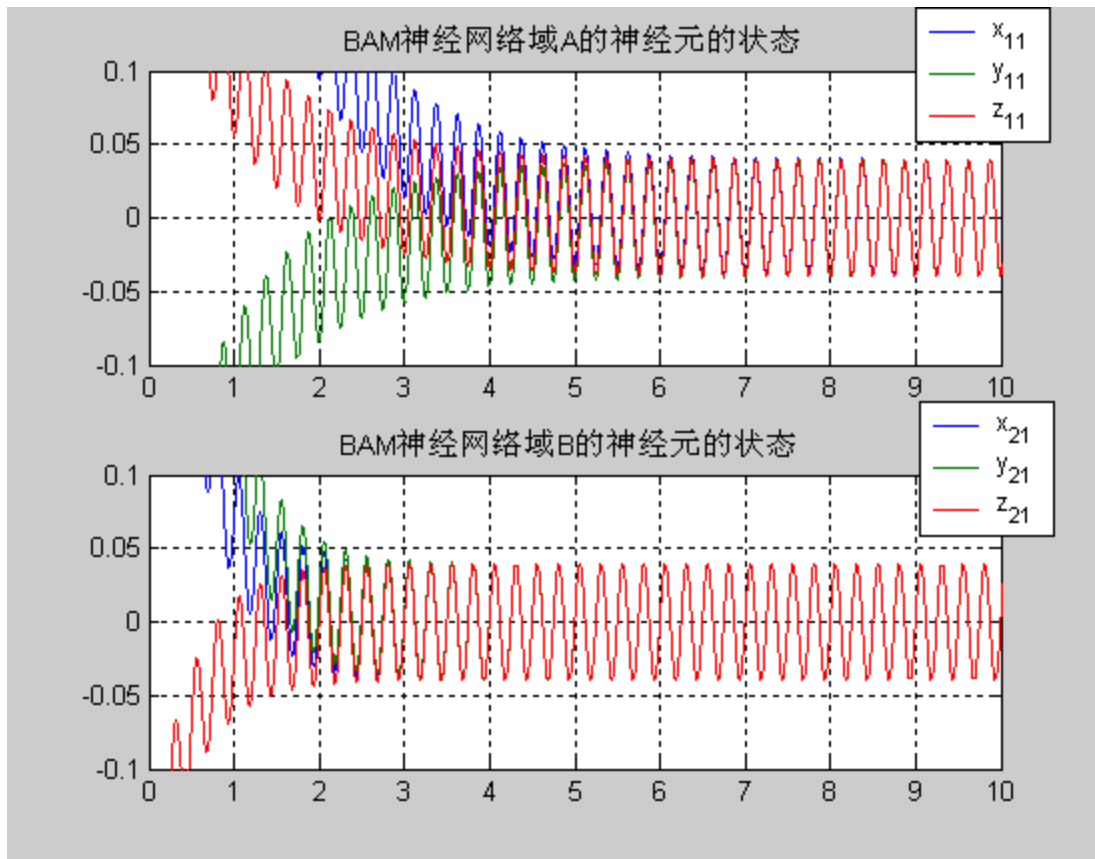
$$x_{11}(n) = 0.6 + 0.2 \cos(2nb\pi), x_{21}(n) = 0.6 + 0.4 \sin(3nb\pi),$$

$$y_{11}(n) = -0.2 + 0.8 \sin(3nb\pi), y_{21}(n) = 0.6 + 0.5 \cos(nb\pi),$$

$$z_{11}(n) = 0.2, z_{21}(n) = -0.2$$

其中 $n \in Z_0^-$

离散神经网络在三种初始条件下的状态图：



五 时滞微分方程解各种数值解法 性能对比分析

1 欧拉法、改进欧拉法和龙格-库塔法性能分析

由仿真结果可以知道，欧拉法是用一条过各点的切线取代曲线来逼近精确解。该算法简单，计算量小，但精度较低；

与欧拉法相比，改进欧拉法是用两个点 (t_n, y_n) 、 (t_{n+1}, y_{n+1}) 的斜率的平均值来确定下一点的 y 值。而且计算时需要和欧拉法结合使用，即用欧拉法对 y_{n+1} 进行预估，再由改进欧拉法计算 y_{n+1} ，精度较高；

龙格-库塔法增加高阶项可提高计算精度，但也同时带来了需要计算高阶导数的困难。算法的关键是利用低阶导数构成的曲线去拟合含有高阶导数的曲线，从而避免了计算高阶导数的问题

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。

对于半离散化法，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，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。

六 结论

1 研究结论：

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。

2 方法论指导意义：

本文所采用的事物性质的对比研究方法和对于数值问题的可视化研究方法们今后研究问题提供了一个非常好的方法。此外，对于处理问题时候，尽可能地运用跨学科的知识来解决现实中的问题，比如本课题就用到了数值分析、常微分方程、数学分析、神经网络和MatLab仿真等多学科知识。

欢迎提问！

谢谢大家！