Journal of System Simulation

Volume 33 | Issue 1 Article 5

1-18-2021

A Quantization-based Integration Method for Stiff Ordinary Differential Equations

Zhihua Li
School of Mechanical Engineering, Hangzhou Dianzi University, Hangzhou 310018, China;

Jiang De
School of Mechanical Engineering, Hangzhou Dianzi University, Hangzhou 310018, China;

Chenjia Wu
School of Mechanical Engineering, Hangzhou Dianzi University, Hangzhou 310018, China;

Zhihua Fan
School of Mechanical Engineering, Hangzhou Dianzi University, Hangzhou 310018, China;

Follow this and additional works at: https://dc-china-simulation.researchcommons.org/journal

 Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

A Quantization-based Integration Method for Stiff Ordinary Differential Equations

Abstract

Abstract: QSS(Quantized State System) has advantages over the traditional time-discrete integration methods in solving general ordinary differential equation (ODE) systems, but it is not suitable for solving the stiff ODE systems. *To effectively solve the stiff ODE systems, a step-correction optimization algorithm based on QSS (SCOA based-on QSS) is proposed, which combines the ideas of the QSS method and the implicit trapezoidal integral method.* The simulation results of three typical stiff ODE cases show that the SCOA based-on QSS algorithm has advantages over other algorithms, and the simulation accuracy can be significantly improved by appropriately reducing the quantum size.

Keywords

ordinary differential equation, stiff system, quantized state system, numerical integration, simulation

Recommended Citation

Li Zhihua, Jiang De, Wu Chenjia, Fan Zhihua. A Quantization-based Integration Method for Stiff Ordinary Differential Equations[J]. Journal of System Simulation, 2021, 33(1): 46-53.

This paper is available in Journal of System Simulation: https://dc-china-simulation.researchcommons.org/journal/vol33/iss1/5

第 33 卷第 1 期 系统仿真学报© Vol. 33 No. 1
2021 年 1 月 Journal of System Simulation Jan. 2021

一种基于量子化状态系统的刚性 ODE 求解方法

李志华，江德，吴晨佳，樊志华
(杭州电子科技大学 机械工程学院，浙江 杭州 310018)

摘要：量子化状态系统(Quantized State System，QSS)在求解一般常微分方程(Ordinary Differential Equation，ODE)系统时，比传统基于时间离散的积分方法更具优势，但 QSS 方法不适合求解刚性 ODE 系统，*为此提出一种基于量子化状态系统的步进校正优化算法(Step-correction Optimization Algorithm Based on QSS，SCOA based-on QSS)，它结合 QSS 方法及隐式算法中梯形积分法的思想，以有效提高刚性 ODE 系统的求解精度和效率*。通过对 3 个典型刚性 ODE 算例的仿真求解，结果表明，SCOA based-on QSS 算法总体上比其他算法更具优势，同时在适当减小量子大小时能显著提高仿真精度。

关键词：常微分方程；刚性系统；量子化状态系统；数值积分；仿真
中图分类号：TP391 文献标志码：A 文章编号：1004-731X (2021) 01-0046-08
DOI: 10.16182/j.issn1004731x.joss.19-0172

A Quantization-based Integration Method for Stiff Ordinary Differential Equations

Li Zhihua, *Jiang De*, *Wu Chenjia*, *Fan Zhihua*
(School of Mechanical Engineering, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract: QSS(Quantized State System) has advantages over the traditional time-discrete integration methods in solving general ordinary differential equation (ODE) systems, but it is not suitable for solving the stiff ODE systems. *To effectively solve the stiff ODE systems, a step-correction optimization algorithm based on QSS (SCOA based-on QSS) is proposed, which combines the ideas of the QSS method and the implicit trapezoidal integral method.* The simulation results of three typical stiff ODE cases show that the SCOA based-on QSS algorithm has advantages over other algorithms, and the simulation accuracy can be significantly improved by appropriately reducing the quantum size.

Keywords: ordinary differential equation; stiff system; quantized state system; numerical integration; simulation

引言

常微分方程(Ordinary Differential Equation，ODE)系统模型主要来自现代机械、电力电子、神经网络、多粒子碰撞动力学等科学与工程领域[1]，许多具有实际意义的 ODE 系统往往同时存在状态变量的“快变”和“慢变”现象，这就是所谓的刚性系统[2]。

ODE 系统仿真求解采用的积分方法基本都是基于时间离散的数值积分方法，如 Euler，Runge-Kutta 等，它们在时间轴上离散模型，逐(时间)点求得状态变量的值，从而推进积分计算。到目前为止，已产生了数百种不同的算法，这些各具特色的算法用于解决不同类型的 ODE 问题[3-4]。

当 ODE 系统为“刚性”时，采用传统时间离散的数值积分方法求解，由于稳定性的原因，需要强

收稿日期：2019-04-23 修回日期：2019-08-09
基金项目：国家重点研发计划“智能机器人”专项(2017YFB1301300)，浙江省自然科学基金(LY18E050008，LY19E050013)
作者简介：李志华(1966-)，男，博士，教授，研究方向为多领域建模与仿真优化、CAD/CAE。E-mail：D98LZH@263.net

Journal of System Simulation, Vol. 33 [2021], Iss. 1, Art. 5

制使用隐式算法，因为所有显式的方法都必须显著减少积分步长以确保数值的稳定性[5]，但无限制的缩短步长容易导致误差的积累从而降低仿真的精度[6]。隐式算法需要在每一个积分步中调用迭代算法来计算下一个值，其过程繁琐且复杂，计算成本随着系统规模的增大而显著增加，仿真效率随之降低，隐式算法不能完全解决精度问题[7-10]。

量子化状态系统(Quantized State System，QSS)方法是一种新的数值积分方法，它与传统基于时间离散的积分方法显著不同，QSS 方法对状态变量进行离散，即系统的状态变量以“量子”为单位跃迁，依次计算每次状态变量跃迁所需要的时间，从而推进积分。这一思想由 Zeigler 等[11]提出，并由 Kofman[12]实现。在求解一般 ODE 系统时，QSS 算法不仅具有强稳定性、高精确度等优点，而且计算过程完全不需要进行迭代，因此仿真效率大大提升[7]。Cellier 等[1,6]后续又提出了 QSS2 和 QSS3，执行二阶与三阶近似，与 QSS1 方法相比，提高了系统仿真的精度[13-16]。国内学者研究较少，杨祎等[7]和李帛洋等[8]在 QSS 算法的基础上先后提出了离散状态事件驱动(Discrete State Event Driven，DSED)方法和后向离散状态事件驱动(Backward DSED，BDSED)方法。

然而，QSS 方法作为一种显式算法，在求解刚性 ODE 系统时会出现仿真数值振荡现象，即不稳定现象[10]，因此，QSS 方法不能完全应用于刚性系统。为了解决这一问题，Kofman 等[2]提出了一种线性隐式量子化状态系统(Linearly Implicit Quantized State System，LIQSS)算法，它可以有效求解一些刚性系统[17]。但是，LIQSS 算法存在一个限制，即要求系统雅可比矩阵的主对角线上有较大元素，否则，仿真轨迹中出现的“假性”振荡会在一定程度上恶化仿真性能[5]。

本文以一般刚性 ODE 系统(即系统雅可比矩阵无特殊要求)为对象，在借鉴 QSS 方法及隐式梯形积分法的基础上，提出一种基于量子化状态系统的步进校正优化算法(Step-correction Optimization Algorithm Based on QSS，SCOA based-on QSS)，有效扩展 QSS 在刚性系统中的应用范围。该算法在 Matlab 平台上予以了实现，对 3 个典型刚性 ODE 算例[18]进行了仿真求解，并与传统数值积分方法、QSS 方法和 LIQSS 算法进行了性能对比。

1 量子化状态系统

QSS 方法是以状态变量的量子化取代传统数值积分的时间离散化。

考虑一组由常微分方程(ODEs)表示的状态方程系统(State Equation System，SES)：

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}(t), t) \tag{1}$$

式中：$\boldsymbol{x}(t) \in R^n$ 为状态向量，QSS 方法将式(1)近似为：

$$\dot{\boldsymbol{x}} = f(\boldsymbol{q}(t), t) \tag{2}$$

式中：$\boldsymbol{q}(t)$为状态变量的量化向量，每一个量化变量 $q_i(t)$通过式(3)的迟滞量化函数得到：

$$q_i(t) = \begin{cases} x_i(t) & \left|x_i(t) - q_i(t^-)\right| \geqslant \Delta Q_i \\ q_i(t^-) & \text{其他} \end{cases} \tag{3}$$

式中：ΔQ_i 为量子，量子是事先给定的，与传统数值积分算法中的时间步长作用类似，量子直接决定了算法误 m 差的大小。从式(3)中可看出 $q_i(t)$遵循分段常数的轨迹，只有当 $x_i(t)$的变化超过某个值(即量子ΔQ_i)，才会引起 $q_i(t)$的变化，称为发生了一次跃迁，且 $q_i(t)$与 $x_i(t)$的差值不会超过一个量子，这也保证了算法的稳定性与全局误差界限[19]。

由于 $q_i(t)$轨迹的特殊性，式(2)的数值解很明确，因此 QSS 算法可以很容易得到仿真。

设 i=1, 2,..., n， j=1, 2,..., n，T 为设置的仿真时间，Δt_{x_i} 为单个变量每一次跃迁所需的时间，Δt 为系统每推进一次仿真所需的时间；此外，变量只有在发生一次跃迁时，即 $\left|q_i - x_i\right| = \Delta Q_i$ 时，才会更新其量化变量 q_i，此时 q_i=x_i；q_i 是否更新决定了变量导数 $\dot{x}_i$ 是否更新；未发生跃迁的状态变量的变化值记为Δx_i；系统仿真推进的时间点记为 t，状态变量在该时间点的值记为 x_i。

QSS 算法的仿真流程图如图 1 所示。

https://dc-china-simulation.researchcommons.org/journal/vol33/iss1/5
DOI: 10.16182/j.issn1004731x.joss.19-0172

第 33 卷第 1 期 系统仿真学报 Vol. 33 No. 1
2021 年 1 月 Journal of System Simulation Jan. 2021

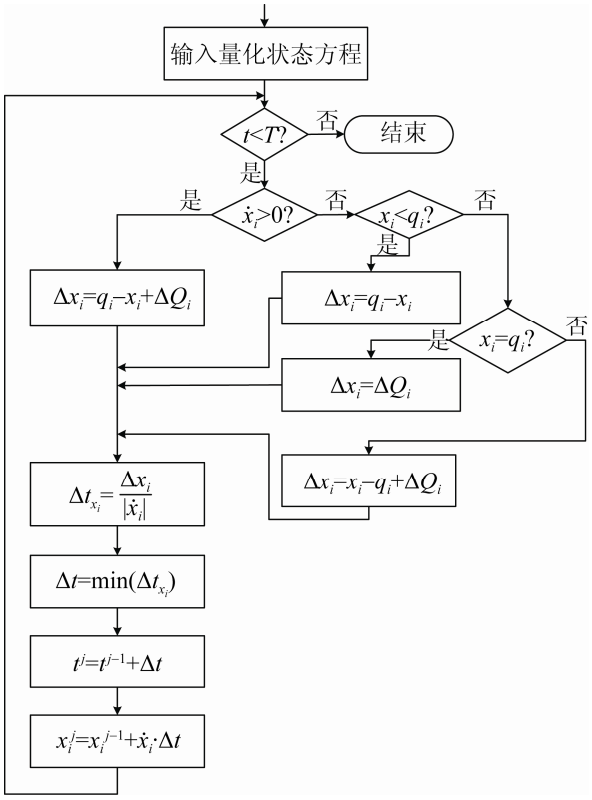

图 1 QSS 算法的仿真流程图
Fig. 1 Simulation flow chart of QSS algorithm

QSS 算法具有很好的全局误差控制性能，这与量子ΔQ_i的取值密不可分，而且 QSS 算法是实时仿真的优异选择，因为算法的异步性，所以算法可以很容易在并行计算机架构上实现[20]。

2 基于量子化状态系统的步进校正优化算法

2.1 SCOA based-on QSS 算法描述

本文提出的 SCOA based-on QSS 算法旨在提高一般刚性 ODE 系统的求解精度和效率，有效扩展 QSS 在刚性系统中的应用范围。该算法结合了 QSS 方法及隐式算法中梯形积分法的思想，梯形法作为隐式算法，能较好保证仿真的稳定性；QSS 方法作为显式算法，不需要进行迭代求解，保证了仿真时间的减少。此外，本算法不再将量化变量值作为系统状态变量的近似值，而是作为预测值，有效地扼制了刚性系统中仿真数值出现的快速振荡，大幅提升仿真效率及仿真精度。

假设系统的量化状态方程如式(2)所示。

令系统状态变量为 x_j，j=1, 2,..., n，则其对应的量化状态变量为 q_j；当在时间点 t 时，量化变量 q_j 的预测值有 2 个，分别为：q_j^+(上限值)、q_j^-(下限值)；计算式分别为：$q_j^+=q_j+\Delta Q_j$，$q_j^-=q_j-\Delta Q_j$，ΔQ_j 为量子；那么系统量化状态变量预测值的导数可分别记为：$\dot{x}_j^+$，$\dot{x}_j^-$，2 个预测值可以根据式(2)求得；仿真推进过程中 x_j 的取值一直趋向 q_j 取值的方向，而 q_j 的取值为：

$$q_j=\begin{cases}q_j^+ & \dot{x}_j^+>0 且 \dot{x}_j^->0\\ q_j^- & \dot{x}_j^+<0 且 \dot{x}_j^-<0\\ \hat{q}_j & \dot{x}_j^+\cdot\dot{x}_j^-<0\end{cases} \tag{4}$$

(1) 仿真推进过程中，如若系统量化状态变量预测值的导数符号一致，即 $\dot{x}_j^+\cdot\dot{x}_j^->0$，则可确定系统变量的运行轨迹，此时根据式(4)可以求得 q_j，然后根据式(2)求得 $\dot{x}_j$ 的值。

这样，状态变量每一次跃迁的时间为：

$$\Delta t_j=\Delta Q_j\Big/\left|\dot{x}_j\right| \tag{5}$$

此时，状态变量的仿真推进时间及变量值计算式分别为：

$$t_j^{(k)}=t_j^{(k-1)}+\Delta t_j \tag{6}$$

$$x_j^{(k)}=x_j^{(k-1)}+\Delta t_j/2\times(\dot{x}_j^{(k-1)}+\dot{x}_j^{(k)}) \tag{7}$$

式中：k 为系统仿真进程中执行的步数。

(2) 当系统量化状态变量预测值的导数符号不同，即 $\dot{x}_j^+\cdot\dot{x}_j^-<0$，则说明仿真进程中状态变量的轨迹发生了变化，那么在这段变量轨迹中必然存在系统量化状态变量预测值的导数为 0，即 $\dot{x}_j=0$；此时：

$$\hat{q}_j=-(u_{jj}/A_{jj}) \tag{8}$$

式中：A_{jj} 为系统雅可比矩阵的对角线元素，$A_{jj}=\partial f/\partial \mathbf{x}$；$u_{jj}=\dot{x}_j-A_{jj}\times q_j$。

根据式(4)计算系统量化状态变量 q_j 的值，即 $q_j=\hat{q}_j$；根据式(2)重置系统状态变量的导数 $\hat{\dot{x}}_j$，此时，变量发生跃迁的时间为：

$$\Delta t_j=\Delta Q_j\Big/\left|\hat{\dot{x}}_j\right| \tag{9}$$

而状态变量的值为：

Published by Journal of System Simulation, 2021

Journal of System Simulation, Vol. 33 [2021], Iss. 1, Art. 5

$$x_j^{(k)} = 1/2 \cdot (x_j^{(k-1)} + \hat{q}_j) \tag{10}$$

无论系统量化状态变量预测值的导数符号正负，整个系统每推进一次仿真的时间为：

$$\Delta t = \min(\Delta t_j) \tag{11}$$

2.2 举例说明

假定 ODE 系统方程为：

$$\begin{cases} \dot{x}_1(t) = 0.01x_2(t) \\ \dot{x}_2(t) = -100x_1(t) - 100x_2(t) + 2020 \end{cases} \tag{12}$$

式中：$x_1(0)=0$，$x_2(0)=20$，$\Delta Q_1=\Delta Q_2=1$。

由于系统的特征值为 $\lambda_1 \approx -0.01$ 和 $\lambda_2 \approx -99.99$，因此系统是刚性的。

将 SCOA 算法运用于式(12)，有：

$$\begin{cases} \dot{x}_1(t) = 0.01q_2(t) \\ \dot{x}_2(t) = -100q_1(t) - 100q_2(t) + 2020 \end{cases} \tag{13}$$

则 SCOA 算法求解步骤如下：

step 1：求出 t=0 时，q_1，q_2 的预测值。

由初始条件可知 q_1、q_2 的估计值分别为 0 与 20。则 q_1，q_2 的预测值分别为 $q_1^+ = 1$，$q_1^- = -1$，$q_2^+ = 19$，$q_2^- = 21$。

step 2：判断系统量化状态变量预测值的导数符号是否相同，即求解 $\dot{x}_j^+$，$\dot{x}_j^-$，然后确定 q_1，q_2 的准确值。

对于 x_1，$\dot{x}_1^+ = 0.01q_2^+ = 0.21 > 0$；$\dot{x}_1^- = 0.01q_2^- = 0.19 > 0$。根据式(4)可得 $q_1 = q_1^+ = 1$。

对于 x_2，$\dot{x}_2^+ = -100q_1 - 100q_2^+ + 2020 = 20 > 0$；$\dot{x}_2^- = -100q_1 - 100q_2^- + 2020 = -180 < 0$。根据式(4)可得 $q_2 = \hat{q}_2$，根据式(8)可解得 $\hat{q}_2 = 19.2$，即 $q_2 = 19.2$。

step 3：确定系统推进时间。

根据式(13)有：$\dot{x}_1 = 0.192$，$\dot{x}_2 = 0$。根据式(5)有，$\Delta t_1 = \Delta Q_1 / \dot{x}_1 = 1/0.192 = 5.208$，$\Delta t_2 = \Delta Q_2 / \dot{x}_2 = 1/0 = \infty$。根据式(11)，求得系统推进时间 $\Delta t = \Delta t_1 = 5.208$。

step 4：求出 $t=t+\Delta t=0+5.208=5.208$ 时，状态变量 x_2 的值。

根据式(10)，可得 $x_2(5.208) = 1/2 \times (x_2(0) + \hat{q}_2) = 1/2 \times (20 + 19.2) = 19.6$。

step 5：求出 t=5.208 时，状态变量 x_1 的值。

根据 step 4 可知 q_2 的估计值为 19.6，则 q_2 的预测值分别为 $q_2^+ = 20.6$，$q_2^- = 18.6$。

根据 step 2~step 3 可得 $\dot{x}_1(5.208) = 0.182$，Δt=5.495。则根据式(7)可得，$x_1(5.208) = x_1(0) + \Delta t/2 \times (\dot{x}_1(0) + \dot{x}_1(5.208)) = 0.974$。

当 $t=t+\Delta t$=5.208+5.495=10.703 时，后面的求解可按照上述过程进行下去。

2.3 SCOA based-on QSS 算法的实现

算法实现(以伪代码的形式)：

```
while(t<T)//设置仿真时间 T
  for 每一个 x_j ∈[1,n]
  until ( |x_j − q_j| = ΔQ_j )
    if ( ẋ_j^+ · ẋ_j^− > 0 ) then
      q_j = x_j ± ΔQ_j
      Δt_j = ΔQ_j / |ẋ_j|   //(这里 Δt_j 为最小跃迁时间)
      x_j^(k) = x_j^(k−1) + Δt_j/2 × (ẋ_j^(k−1) + ẋ_j^(k))
       //(这步借鉴梯形积分法思想)
    else q̂_j = −(u_jj / A_jj)
      Δt_j = ΔQ_j / |x̂̇_j|
      x_j^(k) = 1/2 × (x_j^(k−1) + q̂_j)
       //(对变量的取值进行校正优化)
    end if
  A_jj = (ẋ_j^+ − ẋ_j^−) / (q_j − q_j^−)
  end for
  Δt = min(Δt_j)
   //(系统每推进一步仿真时间取状态变量最小跃迁时间)
  t_j^(k) = t_j^(k−1) + Δt   //(仿真推进时间)
end while
```

3 SCOA based-on QSS 算法仿真验证

3.1 仿真背景

本节引入了 3 个典型刚性 ODE 算例，以 OpenModelica 平台内置的经典求解器 DASSL 求解的值作为基准值[5]，将 SCOA based-on QSS 算法与传统数值积分方法(包括 Euler 法、Runge-Kutta 系列法 ODE45，ODE23s，ODE15s)、QSS 算法和

https://dc-china-simulation.researchcommons.org/journal/vol33/iss1/5
DOI: 10.16182/j.issn1004731x.joss.19-0172

第 33 卷第 1 期 系统仿真学报 Vol. 33 No. 1
2021 年 1 月 Journal of System Simulation Jan. 2021

LIQSS 算法从仿真时间与仿真精度进行性能对比。其中 DASSL 是大多数仿真工具首选的求解器[20]，可用来求解刚性和非刚性 ODE；ODE15s 是 Matlab 平台中求解刚性 ODE 的最典型算法。

(1) SCOA 及 Euler，ODE45，ODE23s，ODE15s，QSS 和 LIQSS 算法全部在 Matlab 平台上实现；

(2) 计算机硬件配置为 Intel i5-5257 处理器，CPU 运行速度为 2.70 GHz，操作系统为 Windows8.1；

(3) 对所有的算例设置相同的误差容限，即 Rel.Tol=Abs.Tol=10^{-3}；

(4) 系统各个状态变量的相对误差计算式为：

$$e_{rr}=\sqrt{\frac{\sum(u_m[k]-u_{\mathrm{DASSL}}[k])^2}{\sum u_{\mathrm{DASSL}}[k]^2}} \tag{14}$$

式中：$u_m[k]$为各算法求的状态变量值；$u_{\mathrm{DASSL}}[k]$为 DASSL 求解器在设定的误差容限很小(10^{-9})的情况下求的状态变量值，在此作为基准值[3]。

3.2 算例验证

算例 1：线性二阶刚性系统[17](见式(12))

算例 2：非线性二阶刚性系统(Van Der Pol 振荡器)[18]

Van Der Pol 振荡器作为一种常见的非线性系统，在非线性动力学的研究中被用来仿真验证非线性动力系统的稳定性。通过状态变量代换的方法，将一般的 Van Der Pol 系统方程化为一阶常微分方程组为：

$$\begin{cases}\dot{x}_1(t)=x_2(t)\\ \dot{x}_2(t)=(1-x_1(t)^2)\cdot x_2(t)\cdot\mu-x_1(t)\end{cases} \tag{15}$$

式中：$\mu=10^{-6}$； $t\in[0,100]$； $x_1(0)=2$； $x_2(0)=0$。

算例 3：非线性三阶刚性系统(The Oregonator 方程，OREGO)[18]

The Oregonator 方程是非常典型的非线性耦合系统，该方程的提出对微分方程的现代求解分析算法的研究作用起了重要作用。

$$\begin{cases}\dot{x}_1(t)=77.27[x_2(t)+x_1(t)\\ \quad(1-8.375\times10^{-6}x_1(t)-x_2(t))]\\ \dot{x}_2(t)=[x_3(t)-x_2(t)(1+x_1(t))]/77.27\\ \dot{x}_3(t)=0.161(x_1(t)-x_3(t))\end{cases} \tag{16}$$

式中：$t\in[0,100]$； $x_1(0)=1$； $x_2(0)=2$； $x_3(0)=3$。

3.3 仿真结果分析

用 SCOA based-on QSS 算法对算例 1 进行仿真求解，得到如图 2 所示的轨迹，x_2 作为“刚性”较大的状态变量并未随着仿真时间的推进而发生振荡。各算法对算例 1 的求解结果如表 1 所示。

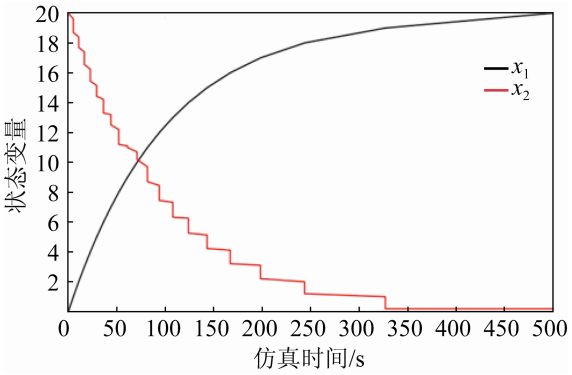

图 2 SCOA based-on QSS 对算例 1 仿真的轨迹

Fig. 2 Trajectory of SCOA based-on QSS simulation for example 1

表 1 算例 1 的仿真结果

Tab. 1 Simulation results of example 1

仿真算法	误差容限	相对误差	CPU 仿真时间/s	仿真步数
Euler	Tol=10^{-3}	3.063×10^{-3}	15.302 706	500 001
ODE45	Tol=10^{-3}	9.483×10^{-2}	11.469 503	60 209
ODE15s	Tol=10^{-3}	7.782×10^{-3}	0.031 776	40
ODE23s	Tol=10^{-3}	9.383×10^{-3}	0.039 272	31
QSS	Tol=10^{-3}, ΔQ=1	—	—	—
LIQSS	Tol=10^{-3}, ΔQ=1	1.033×10^{-4}	0.006 889	46
SCOA	Tol=10^{-3}, ΔQ=1	1.601×10^{-4}	0.012 197	39
	Tol=10^{-3}, ΔQ=0.1	3.395×10^{-5}	0.015 604	77

Published by Journal of System Simulation, 2021

从表 1 可以看出，QSS 算法无解；SCOA 算法无论是在精度(误差)还是效率(CPU 运行时间)上，都远远高于 Euler 法与 ODE45 法，即精度提高了 20 倍，效率提高了 1 000 倍；对比 ODE15s 和 ODE23s，SCOA 算法精度提高了近 50 倍，效率提高了近 3 倍；对比 LIQSS，SCOA 算法略显差些。

用 SCOA based-on QSS 算法对算例 2 进行仿真求解，得到如图 3 所示的轨迹，从图 3 中可以看出，系统变量并未因相互之间的耦合作用而发生振荡。

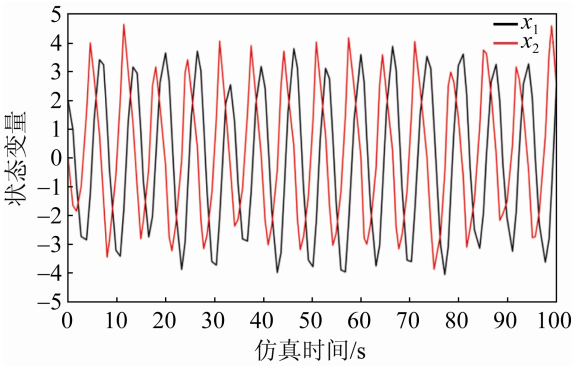

图 3 SCOA based-on QSS 对算例 2 仿真的轨迹
Fig. 3 Trajectory of SCOA based-on QSS simulation for example 2

从表 2 中可以看出，Euler 算法、QSS 算法无解，说明它们对刚性系统的求解不理想；对比 ODE15s 算法，SCOA 算法精度提高了 27 倍，效率提高了 5 倍；对比 LIQSS，SCOA 算法精度提高了 4 倍，效率提高了近 1 倍。

用 SCOA based-on QSS 算法对算例 3 进行仿真求解，得到如图 4 所示的轨迹，从图 4 可以看出，系统变量之间差值非常大(刚性大)，但是变量的仿真轨迹总体平稳，未出现“假性”振荡。从图 4 中的局部放大图可知，在仿真时间达到 60 s 时，变量 x_3 趋于稳定。

从表 3 中可以看出，Euler 算法、QSS 算法及 LIQSS 算法都无解。对比 ODE15s 算法，SCOA 算法在效率略高的情况下，精度提高了近 7 倍。

为了研究量子大小对 SCOA based-on QSS 算法性能的影响，3 个算例都分别选取了 2 个不同的量子值来仿真求解。由表 1~3 可以看出，当减小量子的大小时，CPU 仿真时间相差并不大，但是仿真精度却得到了较大提升。

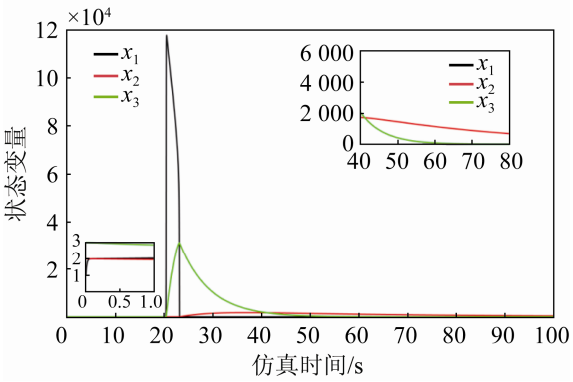

图 4 SCOA based-on QSS 对算例 3 仿真的轨迹
Fig. 4 Trajectory of SCOA based-on QSS simulation for example 3

表 2 算例 2 的仿真结果
Tab. 2 Simulation results of example 2

仿真算法	误差容限	相对误差	CPU 仿真时间/s	仿真步数
Euler	Tol=10^{-3}	—	—	—
ODE45	Tol=10^{-3}	7.015×10^{-2}	644.215 805	3 487 181
ODE15s	Tol=10^{-3}	1.725×10^{-3}	0.392 976	421
ODE23s	Tol=10^{-3}	3.346×10^{-2}	0.375 130	369
QSS	Tol=10^{-3}, ΔQ=0.1	—	—	—
LIQSS	Tol=10^{-3}, ΔQ=0.1	3.092×10^{-4}	0.107 274	599
SCOA	Tol=10^{-3}, ΔQ=0.1	6.143×10^{-5}	0.063 088	329
	Tol=10^{-3}, ΔQ=0.01	1.062×10^{-5}	0.128 551	664

https://dc-china-simulation.researchcommons.org/journal/vol33/iss1/5
DOI: 10.16182/j.issn1004731x.joss.19-0172

第 33 卷第 1 期 系统仿真学报 Vol. 33 No. 1
2021 年 1 月 Journal of System Simulation Jan. 2021

表 3 算例 3 的仿真结果
Tab. 3 Simulation results of example 3

仿真算法	误差容限	相对误差	CPU 仿真时间/s	仿真步数
Euler	Tol=10^{-3}	—	—	—
ODE45	Tol=10^{-3}	9.155×10^{-2}	688.120 867	3 687 017
ODE15s	Tol=10^{-3}	6.254×10^{-3}	1.558 266	243
ODE23s	Tol=10^{-3}	7.908×10^{-2}	0.938 600	300
QSS	Tol=10^{-3}, ΔQ=1	—	—	—
LIQSS	Tol=10^{-3}, ΔQ=1	—	—	—
SCOA	Tol=10^{-3}, ΔQ=1	8.103×10^{-4}	1.463 540	331
	Tol=10^{-3}, ΔQ=0.1	5.527×10^{-5}	1.688 120	352

4 结论

(1) Euler法不适合求解刚性ODE问题；ODE45是显式算法，在仿真求解中为了确保数值的稳定性，必须显著减少积分步长，导致超长的仿真时间，因此很难满足刚性系统在实际运用中的需要。

(2) ODE15s 和 ODE23s 是传统数值积分方法中求解刚性 ODE 问题的较好算法，本文提出的 SCOA 算法在仿真时间与仿真精度比 ODE15s 和 ODE23s 更具优势。

(3) 3 个算例中，QSS 算法都无解，因此它不适合求解刚性 ODE 问题；LIQSS 算法有一个无解，而 SCOA 算法都能仿真求解。因此 SCOA 算法比 QSS 算法和 LIQSS 算法更具优势。

(4) 适当减小量子大小能在几乎不增加仿真时间的同时，有效提高仿真精度。

参考文献：

[1] Cellier E, Kofman E. Continuous System Simulation[M]. New York: Springer, 2006.

[2] Migoni G, Bortolotto M, Kofman E, et al. Linearly Implicit Quantization-based Integration Methods for Stiff Ordinary Differential Equations[J]. Simulation Modelling Practice and Theory (S1569-190X), 2013, 35(6): 118-136.

[3] Hairer E, Wanner G. Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems[M]. Berlin: Springer-Verlag, 1993.

[4] Hairer E, Wanner G. Solving Ordinary Differential Equations I: Nonstiff Problems[M]. Berlin: Springer, 1991.

[5] Dipietro F, Migoni G, Kofman E. Improving a Linearly Implcit Quantized State System Method[C]. Winter Simulation Conference. Virginia: IEEE, 2016: 1084-1095.

[6] Kofman E. A Third Order Discrete Event Simulation Method for Continuous System Simulation[J]. Latin American Applied Research (S0327-0793), 2006, 36(2): 101-108.

[7] Yang Y, Zhao Z M, Tan T, et al. Discrete State Event-driven Simulation and Adaptive Predictive Correction Algorithm[J]. Transactions of China Electrotechnical Society (S1000-6753), 2017, 32(12): 34-41.

[8] 李帛洋, 赵争鸣, 檀添, 等. 后向离散状态事件驱动电力电子仿真方法[J]. 电工技术学报, 2017, 32(12): 43-49.
Li Boyang, Zhao Zhengming, Tan Tian, et al. Simulation Method of Power Electronics Driven by Backward Discrete State Events[J]. Transactions of China Electrotechnical Society, 2017, 32(12): 43-49.

[9] 秦建, 沈沉, 陈颖, 等. 基于量子化状态驱动的时空自律仿真方法[J]. 中国电机工程学报, 2017, 37(10): 2869-2877.
Qin Jian, Shen Chen, Chen Ying, et al. Spatiotemporal Autonomous Simulation Method based on Quantization State Drive[J]. Proceedings of the Chinese Society for Electrical Engineering, 2017, 37(10): 2869-2877.

[10] 檀添, 赵争鸣, 李帛洋, 等. 基于离散状态事件驱动的电力电子瞬态过程仿真方法[J]. 电工技术学报, 2017, 32(13): 42-50.
Tan Tian, Zhao Zhengming, Li Boyang, et al. A Transient Process Simulation Method for Power Electronics based on Discrete State Event-driven[J]. Transactions of China Electrotechnical Society, 2017, 32(13): 42-50.

[11] Zeigier B, Lee J S. Theory of Quantized Systems: Formal Basis for DEVS/HLA Distributed Simulation Environment [J]. In Proceedings of SPIE (S0277-786X), 1998, 3369(1): 49-58.

Published by Journal of System Simulation, 2021

[12] Kofman E, Junco S. Quantized-State-Systems: A DEVS Approach for Continuous System Simulation[J]. Transications of the Society for Modeling and Simulation International (S0740-6797), 2001, 18(1): 2-8.

[13] Fernandez J, Kofman E, Bergero F. A Parallel Quantized State System Solver for ODEs[J]. J. Parallel Distrib. Comput (S0743-7315), 2017, 106(1): 14-30.

[14] Fernández J, Kofman E. A Stand-alone Quantized State System Solver for Continuous System Simulation[J]. Simulation (S0037-5497), 2014, 90(7): 782-799.

[15] Bergero F, Fernández J, Kofman E, et al. Time Discretization Versus State Quantization in the Simulation of a 1D Advection Diffusion Reaction Equation[J]. Simulation (S0037-5497), 2016, 92(1): 47-61.

[16] Chatzivasileiadis S, BonviniI M, Matanza J, et al. Cyber-physical Modeling of Distributed Resources for Distribution System Operations[J]. Proceedings of IEEE (S0018-9219), 2016, 104(4): 789-806.

[17] Migoni G, Kofman E, Cellier F. Quantization-based New Integration Methods for Stiff Ordinary Differential Equations[J]. Simulation (S0037-5497), 2012, 88(4): 118-136.

[18] Weiner R, Kulikow G, Beck S, et al. New Third-and Fourth-order Singly Diagonally Implicit Two-step Peer Triples with Local and Global Error Controls for Solving Stiff Ordinary Differential Equations[J]. Journal of Computational and Applied Mathematics (S0377-0427), 2016, 316(1): 380-391.

[19] Kofman E. Relative Error Control in Quantization Based Integration[J]. Latin American Applied Research (S0327-0793), 2009, 39(3): 231-237.

[20] Federico M, Caselia F, Kofman E, et al. On the Efficiency of Quantization-based Integration Methods for Building Simulation[J]. Building Simulation (S1996-3599), 2017, 11(2): 1-14.

https://dc-china-simulation.researchcommons.org/journal/vol33/iss1/5
DOI: 10.16182/j.issn1004731x.joss.19-0172