Advances in Applied Mathematics 应用数学进展, 2024, 13(5), 2499-2505
Published Online May 2024 in Hans. https://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2024.135238

Hans汉斯

改进的正则化前后扫描迭代算法求解刚性最优控制问题

林钰珩

中国地质大学(武汉)，数学与物理学院，湖北 武汉

收稿日期：2024年4月29日；录用日期：2024年5月22日；发布日期：2024年5月31日

摘　要

最优控制问题广泛应用于工程学、经济学、生物学等众多领域。由于寻求解析解往往极具挑战性，人们通常采用设计合适的数值算法来求解其数值解。在这些问题中，刚性最优控制问题的数值求解方法尤为关键，这类问题的处理通常面临两难选择：问题的刚性特性容易引起数值方法的不稳定性，而过度追求数值格式的稳定性则可能导致计算成本显著增加，使得算法难以实用。本文专注于一类特定的刚性最优控制问题，通过改进传统的正则化前后扫描迭代算法，既保证了算法的稳定性，同时也显著提高了计算效率。最后通过数值实验验证了上述结论。

关键词

最优控制问题，刚性，前后扫描迭代算法

Improved Regularization Forward-Backward Scan Iterative Algorithm for Solving Rigid Optimal Control Problems

Yuheng Lin

School of Mathematics and Physics, China University of Geosciences (Wuhan), Wuhan Hubei

Received: Apr. 29th, 2024; accepted: May 22nd, 2024; published: May 31st, 2024

Abstract

Optimal control problems are prevalent in various fields such as engineering, economics, and biology. As finding analytical solutions is often highly challenging, suitable numerical algorithms are typically employed to derive numerical solutions. Among these, the numerical reso-

文章引用: 林钰珩. 改进的正则化前后扫描迭代算法求解刚性最优控制问题[J]. 应用数学进展, 2024, 13(5): 2499-2505.
DOI: 10.12677/aam.2024.135238

lution of stiff optimal control problems is particularly crucial, as it often involves a dilemma: the stiffness of the problem can easily lead to instability in numerical methods, while excessively prioritizing the stability of numerical schemes can significantly increase computational costs, rendering the algorithms impractical. This paper focuses on a specific type of stiff optimal control problem and enhances the traditional regularized forward-backward scan iterative algorithm. This improvement not only ensures the stability of the algorithm but also significantly enhances its computational efficiency. Finally, the above conclusions are verified by numerical experiments.

Keywords

Optimal Control Problem, Stiffness, Forward-Backward Scan Iterative Algorithm

1. 引言

最优控制问题是推导控制策略的一种数学优化方法，作为变分法的拓展，其历史最早可追溯到三百多年前。其中，带有常微分方程约束的最优控制问题是十分常见且基础性的一类问题，它通常对应着实际生产生活中的控制问题经过数学建模后，再进行空间(或时间)离散后得到的半离散问题，其一般形式表现为：

$$\begin{aligned}&\text{Minimize } J = j\left(x\left(T\right)\right)+\int_0^T L\left(x\left(t\right),u\left(t\right)\right)\mathrm{d}t,\\&\dot{x}\left(t\right)= f\left(x\left(t\right),u\left(t\right)\right),t\in\left[0,T\right]\end{aligned}$$

和一般的最优控制问题类似，带有常微分方程约束的最优控制问题理论研究以苏联学者庞特里亚金和美国学者贝尔曼[1]给出的最优必要条件为基础，在 Hopfield 神经网络优化[2]、混沌优化控制[3]以及鲁棒控制[4]等多个领域都发展出了丰富的理论结果，推动了包括航天航空、物理化学反应精细化研究在内的各个科学研究领域的发展。

然而，绝大多数带常微分方程约束的最优控制问题并不能通过解析的方式求得其最优解，因此有必要构造有效的数值计算方法来求其数值最优解。一般，数值求解最优控制问题的方法分为两大类：直接法和间接法：所谓直接法就是对问题先离散再优化，主要包括直接配点法和控制参数化方法；而间接法就是基于最优性条件对问题先优化，然后再离散求解，具体方法主要包括边界迭代法和拟线性化法。

Li 等人通过引入增广的哈密顿量，构造了一种新的间接法即正则化前后扫描迭代算法用于求解庞特里亚金最小值原理[5]；文献[6]在此基础上证明了采用辛龙格库塔方法离散上述迭代格式的全局收敛性，使得应用正则化前后迭代算法求解带常微分方程约束的最优控制问题成为可能。

在本文中，考虑一类带有刚性微分方程的最优控制问题(OptimalControl Problem, OCP)：

$$\text{Minimize } J = j\left(x\left(T\right)\right)+\int_0^T L\left(x\left(t\right),u\left(t\right)\right)\mathrm{d}t. \tag{1.1}$$

受到微分方程和初值条件约束：

$$\dot{x}\left(t\right)= f\left(x\left(t\right),u\left(t\right)\right)+g\left(x\left(t\right),u\left(t\right)\right),\ t\in\left[0,T\right] \tag{1.2}$$

$$x(0)=x_0, \tag{1.3}$$

其中 f 为非刚性项，而 g 为刚性项：在对现实世界系统的数学建模中，f 和 g 通常是对系统的空间导数利用有限差分或有限元方法近似后剩下的时间导数算子，由这两个算子引起的时间尺度可能会有很大的不同；尽管整个方程依然是刚性的，但在选代离散求解的过程中对于 f 和 g 分别进行不同处理是非常有意义的：如果为了保证稳定性而对包括非刚性项之内的整个系统采用隐式格式进行求解，将会产生十分昂贵的计算成本，但这在一定程度上是可以避免的。

2. 改进算法的构造

2.1. 正则化前后扫描迭代算法

引入协态变量 λ，定义上述问题的哈密顿量如下：

$$H(x,\lambda,u):=L(x,u)+\lambda^{\mathrm{T}}\left(f(x,u)+g(x,u)\right)$$

在适当条件下，最优控制问题(1.1)~(1.3)的一阶最优性条件如下[7] [8]：

$$\dot{x}=H_{\lambda}(x,\lambda,u)=f(x,u)+g(x,u),\ x(0)=x_0, \tag{2.1}$$

$$\lambda=-H_x(x,\lambda,u)=-L_x(x,u)-f_x(x,u)^{\mathrm{T}}\lambda-g_x(x,u)^{\mathrm{T}}\lambda,\ \lambda(T)=j'(x(T)), \tag{2.2}$$

$$0=H_u(x,\lambda,u)=L_u(x,u)+f_u(x,u)^{\mathrm{T}}\lambda+b_u(x,u)^{\mathrm{T}}\lambda, \tag{2.3}$$

Li 等人在 2018 年提出了一种正则化前后扫描迭代的算法，来求解一般最优控制问题中的庞特里亚金最大值原理(即一阶最优性条件)，该方法借鉴了增广拉格朗日函数的思想，引入拓展的哈密顿函数：

$$\tilde{H}(x,\lambda,u,p,q)=H(x,\lambda,u)-\frac{\rho}{2}\left(\left\|p-H_{\lambda}(x,\lambda,u)\right\|^2+\left\|q+H_x(x,\lambda,u)\right\|^2\right)$$

正则化前后扫描迭代算法可以描述为：

算法 1 正则化前后扫描迭代算法(Extended MSA)

1：初始化：$u_0\in\mathcal{U},\rho>0$；

2：**for *k* = 0,1,⋯do**

3：　　计算 $\dot{x}^{(k+1)}=\tilde{H}_{\lambda}\left(x^{(k+1)},\lambda^{(k)},u^{(k)},\dot{x}^{(k+1)},\dot{\lambda}^{(k)}\right)$；

4：　　计算 $\dot{\lambda}^{(k+1)}=-\tilde{H}_x\left(x^{(k+1)},\lambda^{(k+1)},u^{(k)},\dot{x}^{(k+1)},\dot{\lambda}^{(k+1)}\right)$；

5：　　计算 $u^{(k+1)}=\arg\max_{u(t)\in\mathcal{U}}\tilde{H}\left(x^{(k+1)},\lambda^{(k+1)},u,\dot{x}^{(k+1)},\dot{\lambda}^{(k+1)}\right)$；

6：**end for**

其中正则化参数 $\rho>0$。需要注意到，在算法的第三、四步分别更新 x 和 λ 时，拓展的哈密顿函数都等同于原本的哈密顿函数，正则项仅在第五步更新 u 时起作用。

要想得到最优解，还需要在算法的框架下采用某种离散格式，将连续最优控制问题进行离散选代求解。在离散格式的选择上，为了确保计算结果的收敛性，将采用辛龙格库塔方法[6]，完整的离散格式将在下一节中介绍。

2.2. 改进后的正则化前后扫描迭代算法

为了方便后续对 f 和 g 进行不同处理，将状态变量 x 分裂成两个不同的状态变量 x_1 和 x_2，重新定义

问题的哈密顿量：

$$K\left(x_1,x_2,\lambda,u\right):=L\left(x_2,u\right)+\lambda^{\mathrm{T}}\left(f\left(x_1,u\right)+g\left(x_2,u\right)\right)$$

需要注意到，当 x_1,x_2 取值相同时，新定义的哈密顿量 K 与之前的哈密顿量 H 相同，即

$$K\left(x_1,x_2,\lambda,u\right)=H\left(x,\lambda,u\right),\ x_1=x_2=x.$$

在此基础上，依据正则化前后扫描迭代算法的构造思路，引入正则化参数 ρ，定义拓展的哈密顿量：

$$\begin{aligned}\tilde{K}\left(x_1,x_2,\lambda,u,p,q\right)&=K\left(x_1,x_2,\lambda,u\right)\\&\quad-\frac{\rho}{2}\left(\left\|p-K_{\lambda}\left(x_1,x_2,\lambda,u\right)\right\|^2+\left\|q+K_{x1}\left(x_1,x_2,\lambda,u\right)+K_{x2}\left(x_1,x_2,\lambda,u\right)\right\|^2\right)\end{aligned}\tag{2.4}$$

在选代更新的过程中为了降低计算成本，对于 f 这一项采用显式格式进行更新计算，由于算子 f 是非刚性的，所以显式更新也不会影响其数值稳定性；对于 f 以外的其他项，由于其具有刚性，为保证数值算法的稳定性，依然采用隐式更新。具体而言，在第 $k+1$ 次迭代时，根据以下格式按顺序更新 x、λ 和 u：

$$\dot{x}^{(k+1)}=\tilde{K}_{\lambda}\left(x^{(k)},x^{(k+1)},\lambda^{(k)},u^{(k)},\dot{x}^{(k+1)},\dot{\lambda}^{(k+1)}\right)\tag{2.5}$$

$$\begin{aligned}\dot{\lambda}^{(k+1)}&=-\tilde{K}_{x_1}\left(x^{(k+1)},x^{(k+1)},\lambda^{(k+1)},u^{(k)},\dot{x}^{(k+1)},\dot{\lambda}^{(k+1)}\right)\\&\quad-\tilde{K}_{x_2}\left(x^{(k+1)},x^{(k+1)},\lambda^{(k+1)},u^{(k)},\dot{x}^{(k+1)},\dot{\lambda}^{(k+1)}\right)\end{aligned}\tag{2.6}$$

$$\dot{u}^{(k+1)}=\arg\max_{u(t)\in\mathcal{U}}\tilde{K}\left(x^{(k+1)},x^{(k+1)},\lambda^{(k+1)},u^{(k+1)},\dot{x}^{(k+1)},\dot{\lambda}^{(k+1)}\right)\tag{2.7}$$

上式经由 s 阶辛龙格库塔格式离散后的迭代格式如下：

$$x_{n+1}^{(k+1)}=x_n^{(k+1)}+\tau\sum_{i=1}^{s}b_i\tilde{K}_{\lambda}\left(X_{n,i}^{(k)},X_{n,i}^{(k+1)},\Lambda_{n,i}^{(k)},U_{n,i}^{(k)},\frac{x_{n+1}^{(k+1)}-x_n^{(k+1)}}{\tau},\frac{\lambda_{n+1}^{(k)}-\lambda_n^{(k)}}{\tau}\right)\tag{2.8}$$

$$X_{n,i}^{(k+1)}=x_n^{(k+1)}+\tau\sum_{j=1}^{s}a_{ij}\tilde{K}_{\lambda}\left(X_{n,i}^{(k)},X_{n,i}^{(k+1)},\Lambda_{n,i}^{(k)},U_{n,i}^{(k)},\frac{x_{n+1}^{(k+1)}-x_n^{(k+1)}}{\tau},\frac{\lambda_{n+1}^{(k)}-\lambda_n^{(k)}}{\tau}\right)\tag{2.9}$$

$$\begin{aligned}\lambda_{n+1}^{(k+1)}&=\lambda_n^{(k+1)}-\tau\sum_{i=1}^{s}b_i\left[\tilde{K}_{x_1}\left(X_{n,i}^{(k+1)},X_{n,i}^{(k+1)},\Lambda_{n,i}^{(k+1)},U_{n,i}^{(k)},V_{n,i}^{(k)},\frac{x_{n+1}^{(k+1)}-x_n^{(k+1)}}{\tau},\frac{\lambda_{n+1}^{(k+1)}-\lambda_n^{(k+1)}}{\tau}\right)\right.\\&\quad\left.+\tilde{K}_{x_2}\left(X_{n,j}^{(k+1)},X_{n,j}^{(k+1)},\Lambda_{n,j}^{(k+1)},U_{n,j}^{(k)},\frac{x_{n+1}^{(k+1)}-x_n^{(k+1)}}{\tau},\frac{\lambda_{n+1}^{(k+1)}-\lambda_n^{(k+1)}}{\tau}\right)\right]\end{aligned}\tag{2.10}$$

$$\begin{aligned}\lambda_{n+1}^{(k+1)}&=\lambda_n^{(k+1)}-\tau\sum_{j=1}^{s}\tilde{a}_{ij}\left[\tilde{K}_{x_1}\left(X_{n,j}^{(k+1)},X_{n,j}^{(k+1)},\Lambda_{n,j}^{(k+1)},U_{n,j}^{(k)},\frac{x_{n+1}^{(k+1)}-x_n^{(k+1)}}{\tau},\frac{\lambda_{n+1}^{(k+1)}-\lambda_n^{(k+1)}}{\tau}\right)\right.\\&\quad\left.+\tilde{K}_{x_2}\left(X_{n,j}^{(k+1)},X_{n,j}^{(k+1)},\Lambda_{n,j}^{(k+1)},U_{n,j}^{(k)},\frac{x_{n+1}^{(k+1)}-x_n^{(k+1)}}{\tau},\frac{\lambda_{n+1}^{(k+1)}-\lambda_n^{(k+1)}}{\tau}\right)\right]\end{aligned}\tag{2.11}$$

$$u_{n+1}^{(k+1)}=\sum_{i=1}^{s}b_iU_{n,i}^{(k+1)}\tag{2.12}$$

$$0=\tilde{K}_U\left(X_{n,i}^{(k+1)},x_{n,i}^{(k+1)},\Lambda_{n,i}^{(k+1)},U_{n,i}^{(k+1)},\frac{x_{n+1}^{(k+1)}-X_n^{(k+1)}}{\tau},\frac{\lambda_{n+1}^{(k+1)}-\lambda_n^{(k+1)}}{\tau}\right) \tag{2.13}$$

其中 $X_n=\left(X_{n,1},\cdots,X_{n,s}\right),\Lambda_n=\left(\Lambda_{n,1},\cdots,\Lambda_{n,s}\right),U_n=\left(U_{n,1},\cdots,U_{n,s}\right)$ 分别表示状态变量 x、协态变量 λ 和控制变量 u 在辛龙格库塔格式下在第 $n+1$ 个小区间中内点上的取值。另外，为了保证离散格式的辛性，上述公式中的系数除了要满足龙格库塔方法的基本要求外，还需满足以下关系[9]：

$$\tilde{a}_{ij}=b_j-b_ja_{ij}/b_i$$

需要注意的是，与寻常的正则化前后扫描迭代算法相比，改进后的算法只在状态变量更新过程中有所不同，在协态变量和控制变量的更新计算中，状态变量为已知量，不会发生改变。

3. 数值实验

为了探讨新构造出来的算法的性质，本章采用 Python3.9 编程实现该算法来求解一个刚性最优控制问题实例。考虑如下带有奇异扰动微分方程约束的刚性最优控制问题

$$\text{Minimize } J=c(1)$$

受微分方程和初值条件约束：

$$c(t)=\frac{1}{2}\left(u^2(t)+x^2(t)+4z^2(t)\right),\ c(0)=0, \tag{3.1}$$

$$\dot{x}(t)=z(t)+u(t),\ x(0)=1, \tag{3.2}$$

$$\dot{z}(t)=\frac{1}{\epsilon}\left(\frac{1}{2}x(t)-z(t)\right),\ z(0)=\frac{1}{2}, \tag{3.3}$$

其中 $\epsilon>0$ (在本实验中不失普遍性地取 $\epsilon=0.15$)。显然，微分方程中的刚性项和非刚性项可以分开来独立表示，即令：

$$y=\begin{pmatrix}c\\x\\z\end{pmatrix},\ f(y,u)=\begin{pmatrix}\frac{1}{2}\left(u^2+x^2+4z^2\right)\\z+u\\0\end{pmatrix},\ g(y)=\frac{1}{f}\begin{pmatrix}0\\0\\\frac{1}{2}x-z\end{pmatrix}, \tag{3.4}$$

这样就将非刚性项 f 和刚性项 y 分离开了。

设步长 $\tau=0.01$，选代终止条件 $\delta=10^{-5}$，离散格式选取辛中点格式 $\left(s=1,a_{11}=b_1=0.5\right)$。经过 118 次迭代，得到问题数值解(见图 1)。

随着步长 τ 的变小，指标函数也趋向收敛(见图 2)。

与改进前的算法相比，在相同的选代终止条件下，新算法所需的迭代次数更少，即新算法的数值收敛速度更快，并且，这个速度上的差距会随着步长的变大而增大(见图 3)。

同样地，与改进前的算法算法相比，新算法拥有更少的 CPU 耗时，这表明改进后的新算法有着更高的计算效率(见图 4)。

虽然在本实验的示例问题求解过程中两种算法的 CPU 耗时差距不大，但当系统变得更加高维或体量更大时，采用新算法将能节约相当可观的时间，降低计算成本。因此，本文中对传统的正则化前后扫描迭代算法的改进是有效且有意义的。

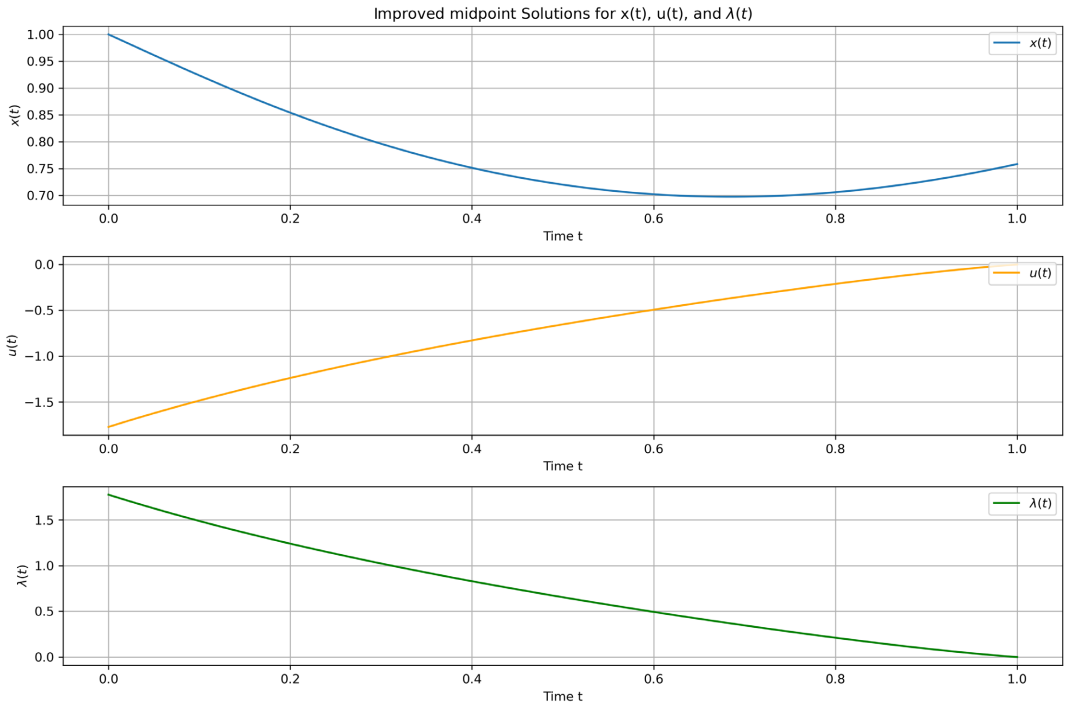

Figure 1. Improved midpoint Solutions for $x(t)$, $u(t)$, and $\lambda(t)$
图 1. 用改进后中点方法求得的 $x(t)$，$u(t)$和 $\lambda(t)$

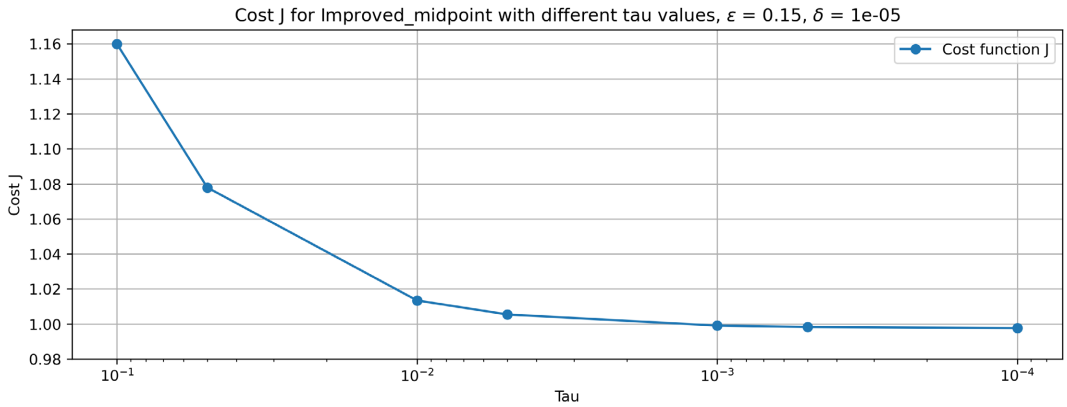

Figure 2. Cost J for improved midpoint with different tau values
图 2. 改进后方法在不同 tau 值下得到的成本泛函值

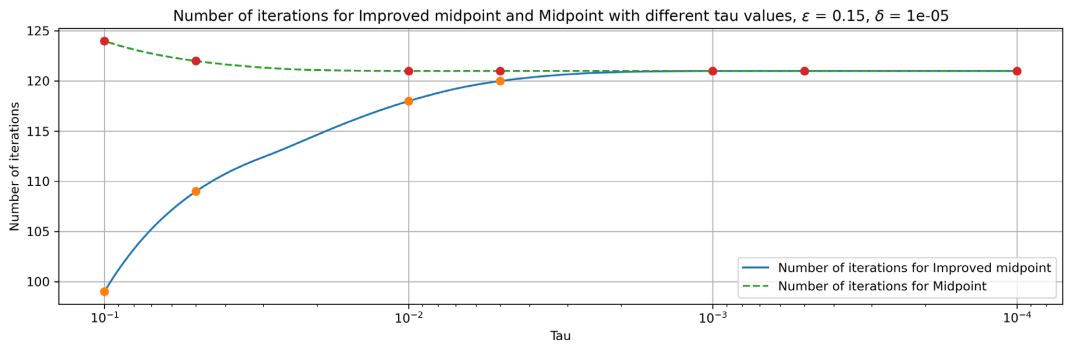

Figure 3. Number of iterations for improved midpoint and Midpoint with different tau values
图 3. 改进后中点法和原中点法在不同 tau 取值下所需的迭代次数

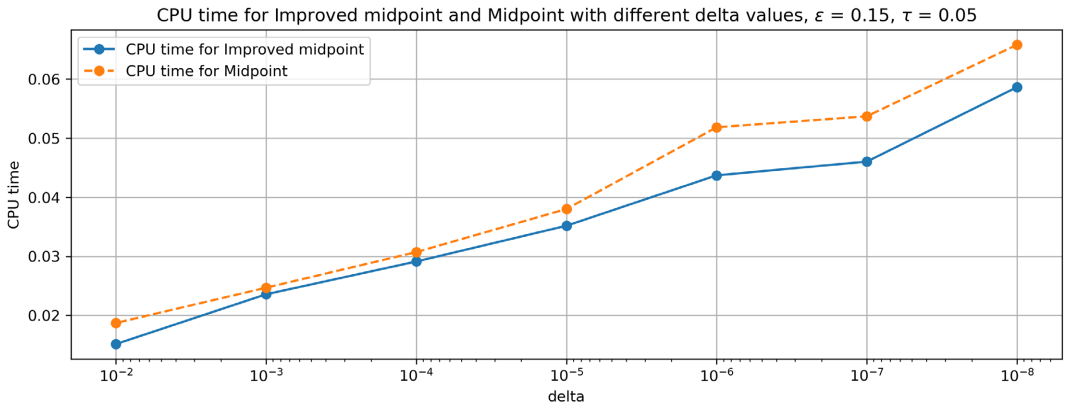

Figure 4. CPU time for improved midpoint and Midpoint with different delta values
图 4. 改进后中点法和原中点法在不同 delta 取值下所需的 CPU 耗时

4. 总结与展望

本文向经典的正则化前后扫描迭代算法迭代过程中引入一种混合更新机制，对非刚性项项采用显式更新，而对其余项则采用隐式更新，使其用于一类特殊的有实际应用背景的刚性最优控制问题时，相比经典的算法有着更高的计算效率和更低的计算成本，最后通过数值实验证实了这个结论。

参考文献

[1] Halkin, H. (1964) Optimal Control for Systems Described by Difference Equations. *Advances in Control Systems*, **1**, 173-196. https://doi.org/10.1016/B978-1-4831-6717-6.50009-7

[2] Hopfield, J.J. and Tank, D.W. (1985) "Neural" Computation of Decisions in Optimization Problems. *Biological Cybernetics*, **52**, 141-152. https://doi.org/10.1007/BF00339943

[3] Yang, D., Li, G. and Cheng, G. (2007) On the Efficiency of Chaos Optimization Algorithms for Global Optimization. *Chaos, Solitons & Fractals*, **34**, 1366-1375. https://doi.org/10.1016/j.chaos.2006.04.057

[4] Lin, F. (2007) Robust Control Design: An Optimal Control Approach. John Wiley & Sons, Hoboken. https://doi.org/10.1002/9780470059579

[5] Li, Q., Chen, L., Tai, C., *et al.* (2017) Maximum Principle Based Algorithms for Deep Learning. *Journal of Machine Learning Research*, **18**, 5998-6026.

[6] Liu, X. and Frank, J. (2021) Symplectic Runge-Kutta Discretization of a Regularized Forward-Backward Sweep Iteration for Optimal Control Problems. *Journal of Computational and Applied Mathematics*, **383**, Article ID: 113133. https://doi.org/10.1016/j.cam.2020.113133

[7] Liberzon, D. (2011) Calculus of Variations and Optimal Control Theory: A Concise Introduction. Princeton University Press, Princeton. https://doi.org/10.2307/j.ctvcm4g0s

[8] Troutman, J.L. (2012) Variational Calculus and Optimal Control: Optimization with Elementary Convexity. Springer Science & Business Media, Berlin.

[9] Sanz-Serna, J.M. (2016) Symplectic Runge-Kutta Schemes for Adjoint Equations, Automatic Differentiation, Optimal Control, and More. *SIAM Review*, **58**, 3-33. https://doi.org/10.1137/151002769