SCIENTIA SINICA Mathematica

论文



线性常微分方程的全局误差估计和优化求解方法

献给张景中、杨路教授85华诞

杨文强1,2, 吴文渊1*

- 1. 中国科学院重庆绿色智能技术研究院自动推理与认知重庆市重点实验室, 重庆 400714;
- 2. 中国科学院大学重庆学院, 重庆 400714

E-mail: yangwenqiang@cigit.ac.cn, wuwenyuan@cigit.ac.cn

收稿日期: 2019-12-20; 接受日期: 2020-10-20; 网络出版日期: 2020-12-22; * 通信作者中国科学院前沿科学重点项目 (批准号: QYZDB-SSW-SYS026)、重庆市科技创新引导专项 (批准号: cstc2018jcyj-yszxX0002)、国家自然科学基金 (批准号: 11671377 和 11771421) 和中国科学院"西部之光"资助项目

摘要 线性常微分方程初值问题求解在许多应用中起着重要作用.目前,已存在很多的数值方法和求解器用于计算离散网格点上的近似解,但很少有对全局误差 (global error) 进行估计和优化的方法.本文首先通过将离散数值解插值成为可微函数用来定义方程的残差; 再给出残差与近似解的关系定理并推导出全局误差的上界; 然后以最小化残差的二范数为目标将方程求解问题转化为优化求解问题; 最后通过分析导出矩阵的结构, 提出利用共轭梯度法对其进行求解. 之后将该方法应用于滤波电路和汽车悬架系统等实际问题.实验分析表明,本文估计方法对线性常微分方程的初值问题的全局误差具有比较好的估计效果,优化求解方法能够在不增加网格点的情形下求解出线性常微分方程在插值解空间中的全局最优解.

关键词 线性常微分方程 全局误差估计 残差 优化求解

MSC (2010) 主题分类 34A30, 34B07, 34D23, 34H05

1 引言

微分方程 (differential equation, DE) 作为描述事物运动和发展规律最基本的工具之一,在工程力学、控制工程、电力电子技术和化学反应过程等多个领域都有着十分重要的应用. 绝大多数系统都是非线性的复杂系统,在应用层面上对其进行研究存在着计算成本大和实时性差等缺点,再加上现有的线性化手段在大多数情形下都能够很好地描述非线性系统在局部的特性,特别是短时控制系统,因而更加深入地研究线性微分方程系统显得尤为必要. 然而,大多数微分方程,即使是线性时变的,也难以获得解析解. 我们通常采用数值方法求解得到离散的数值解,然后额外利用插值方法来获得连续的

英文引用格式: Yang W Q, Wu W Y. Global error estimation for linear ordinary differential equations and their numerical optimal solutions (in Chinese). Sci Sin Math, 2021, 51: 239–256, doi: 10.1360/SSM-2019-0313

近似解, 进而对连续的微分方程进行分析描述, 但该方法的困难在于很难给出满足全局精度要求的近似解.

在安全攸关的控制系统中,采用微分方程进行描述和建模非常普遍也尤其重要.如欧洲列车运行控制协议和飞机避碰协议、两卫星实现正常的交会对接等状态切换控制正确性的验证 (参见文献 [1-3]),整个状态切换控制的过程中准确性显得尤为重要,某些状态切换控制需要考虑系统的连续状态而不仅仅是网格点上的状态,因此要求该系统微分方程求解的全局误差可控,这关系到系统的安全性.所以,全局误差小的近似解以及对近似解的误差估计是保障系统安全性的重要手段.

本文试图从误差估计和最优解两个方面回答在线性常微分方程 (ordinary differential equation, ODE) 初值问题中如何确定解的全局误差上界和提高解的质量这两个问题.

线性常微分方程初值问题的标准形式如下:

$$x' = A(t) \cdot x(t) + q(t),$$

$$x(t_0) = x_0,$$
(1.1)

其中 $\mathbf{x} = \mathbf{x}(t) : \mathbb{R} \to \mathbb{R}^n$ 是关于时间的一个向量函数, 矩阵 $\mathbf{A}(t)$ 是 $n \times n$ 阶状态矩阵, n 维列向量 $\mathbf{q}(t)$ 是系统的非齐次项, $\mathbf{x}(t_0) = \mathbf{x}_0$ 为在初始时刻 t_0 的初值条件, $\mathbf{x}_0 \in \mathbb{R}^n$, t 为时间.

在机械动力学和控制工程等领域的模拟仿真中,系统在某时刻后的状态变化,通常被描述为线性常微分方程的初值问题. 如果常微分方程在时间 t 上连续,在 x 上 Lipschitz 连续,那么它在边界条件的领域范围内一定存在唯一的精确解 x^* . 目前,在线性常微分方程的初值问题 [4] 和边界值问题 [5,6] 上,已有非常成熟的数值求解方法和 p 阶误差的分析方法,并且开发了很多面向应用的求解器,如MATLAB [7] 和 Maple [8]. 但如何严格给出由求解器数值解插值得到的近似解与精确解的距离上界,以及如何在数值解的基础上提高近似解的质量,就我们所知,这方面的工作还比较少.

对于误差估计而言,通过残差对其进行界定具有许多优点,最大的优点就是它能够用来评估整个求解区间内的误差,而不仅仅是每个网格格点上的局部误差 $^{[9]}$. 研究人员通常喜欢将求得的数值解代回原方程求出残差 $^{[10-12]}$,再进行误差估计. Constantinescu 在文献 $^{[13]}$ 中利用时间步进策略结合残差给出了一个 p 你"全局误差"的估计方法,能够更好地提高误差估计的准确性,但本质上只是在格点上的误差估计,而且舍去了高阶项,不能说是准确的误差估计. Enright $^{[14]}$ 提出了合理的估计残差范数的方法,但没有考虑每个点导数的数值误差,这些导数的不准确进一步导致残差定义的不准确,从而导致误差估计的不准确.另一些文献 $^{[10,12,15]}$ 在考虑了导数问题的误差估计方面做了大量工作,但其中大部分都在努力解决局部误差问题.

一般情形下,我们无法求出常微分方程的精确解,但幸运的是,我们可以通过很多的数值方法求解器求得数值解,如 MATLAB等,进一步可以减小步长来逼近精确解,基于此,很多人在这方面做出了重要的研究工作(参见文献 [16–18]). 然而,无论步长取得多小,局部误差控制得多好,我们仍然无法回答数值解与精确解之间到底相差多少这个问题.全局误差估计方法 [19–22] 能够部分地回答这个问题,但是这些方法计算量太大而且条件过于苛刻.文献 [23] 提出利用 Lipschitz 常数来协助误差上界的估计,给出了误差估计上界

$$|\Delta x|_{\text{max}} \leqslant h \cdot \frac{M}{2L} \cdot (e^{Lt} - 1),$$
 (1.2)

其中 h 为步长, M 为 $\|x''\|$ 的估计上界, L 为 Lipschitz 常数, 但由于 Lipschitz 常数为正, 导致估计上界随时间增长呈指数增长, 且增长速率过大, 不具有实用性. 文献 [11] 通过推导条件数与残差的关系估计全局误差, 但求解条件数需要对齐次常微分方程的基本解 (fundamental solution) 进行求解, 在多

数情形下这一点几乎无法实现. 文献 [24] 专门针对稳定系统的常微分方程提出利用 Lyapunov 定理构造全局误差控制方程来限制误差上限, 能够起到误差估计和提高解的精度的作用, 但构造全局误差控制方程也需要基本解的求解. 文献 [25] 提出了一种区间求解的方法, 仅适用于初值问题求解, 该方法基于 Taylor 级数展开, 估计的范围和计算精度取决于级数展开的项数, 精度提高会造成计算量急剧增大; 同时, 经实验验证, 区间求解方法需要大量的预处理时间; 在相同的步长和阶数的情形下, 区间求解方法的精度要低于 Runge-Kutta 等数值求解算法. 因此, 在无法获取精确解的情形下, 衡量常微分方程数值解的可靠性仍然是一个重要的研究问题.

在数值求解中,解的精度往往取决于步长,但更小的步长往往意味着更大的计算量,如何在固定步长的情形下减小全局误差是一个很有难度的问题,目前还没有太多相关的研究."缺陷修正"(defect correction)方法^[26,27],类似于 Newton 迭代方法求解非线性方程组,将残差代回方程进行迭代数值求解,可用于非线性常微分方程,但它们都只是考虑了在网格点上的局部误差.不同于文献 [11] 中所述的在给定数值解的情形下寻找最优的插值点,本文主要研究在求解器给出数值解网格点后,采用插值法构造一个解空间,然后在这个解空间中寻找全局误差最优的解.

需要说明的是,本文若无特别的说明,所针对的问题都是初值问题.

2 预备知识

2.1 Hermite 插值法

对于一阶连续的微分方程, 我们采用 Hermite 三次样条插值来构造微分方程的一个近似解 (参见文献 [11,28]).

定理 2.1 设 $\{t_0, ..., t_m\}$ 为 $[t_0, t_m]$ 区间内的网格点集合, 其中 $t_0 < t_1 < ... < t_m, m \ge 2$. 对任意的 i = 0, ..., m - 1, 在区间 $[t_i, t_{i+1}]$ 上存在唯一的三次样条曲线 $\tilde{x} \in C^1[t_0, t_m]$ 都满足

$$\widetilde{x}(t_i) = x_i, \quad \widetilde{x}'(t_i) = x_i'.$$

令 $h_i = t_{i+1} - t_i$, $\tau = (t - t_i)/h_i$, 其中 $\tau \in [0,1]$, 那么该曲线 \tilde{x} 形式如下:

$$\widetilde{x}_i(\tau) = \alpha_0(\tau) \cdot x_i + h_i \alpha_1(\tau) \cdot x_i' + \beta_0(\tau) \cdot x_{i+1} + h_i \beta_1(\tau) \cdot x_{i+1}',$$

其中

$$\alpha_0(\tau) = 2\tau^3 - 3\tau^2 + 1$$
, $\alpha_1(\tau) = \tau^3 - 2\tau^2 + \tau$, $\beta_0(\tau) = -2\tau^3 + 3\tau^2$, $\beta_1(\tau) = \tau^3 - \tau^2$.

插值方法的选择在实践中非常重要,因为它常常导致不同的应用结果.这里使用一个简单的插值来演示这个想法,更多复杂的插值应用方法可参见文献[29-31].

定理 2.2 [32] 设 $f \in C^4[a,b]$, 令曲线 p 为 f 在区间 [a,b] 上的 Hermite 三次样条插值, m+1 个 网络点满足 $a=x_0 < x_1 < \cdots < x_m = b$, 那么该插值区间的余项误差满足以下界限:

$$||f - p||_{\infty} \le \frac{1}{384} h^4 ||f^{(4)}||_{\infty},$$

其中 $f^{(4)}$ 是 f 关于自变量 x 的四阶导数, 最大步长 $h = \max_i (x_i - x_{i-1}), \|\cdot\|_{\infty}$ 定义为在 [a, b] 区间上的无穷范数.

定义 2.1 设 $\{x_0, x_1, \dots, x_m\}$ 和它们的导数 $\{x'_0, x'_1, \dots, x'_m\}$ 为线性常微分方程 (1.1) 的数值解, 其中 $t \in [t_0, t_m]$, t 存在唯一的 n 维三次样条曲线 $\tilde{x}(t) \in C^1[t_0, t_m]$, 在每一个时间格点 t_k 上都满足

$$\widetilde{\boldsymbol{x}}(t_k) = \boldsymbol{x}_k, \quad \widetilde{\boldsymbol{x}}'(t_k) = \boldsymbol{x}_k'.$$

2.2 残差的定义

为了评估解的精度, 我们将继续沿用文献 [11] 提出的残差定义, 进一步提出了残差二范数和全局误差的定义, 其他更详细的误差分析的研究可以进一步参考该文献.

定义 2.2 设 n 维向量 $\tilde{x}(t) \in C^1[t_0, t_m]$ 是线性常微分方程 (1.1) 的近似解, 那么它的残差为

$$\delta(t) = \widetilde{x}'(t) - A(t) \cdot \widetilde{x}(t) - q(t), \tag{2.1}$$

残差的二范数为

$$\sqrt{\int_{t_0}^{t_m} \boldsymbol{\delta}(t)^{\mathrm{T}} \boldsymbol{\delta}(t) dt}.$$

设n维向量 $x^*(t)$ 为线性常微分方程(1.1)的精确解,满足

$$(\boldsymbol{x}^*)' = \boldsymbol{A}(t) \cdot \boldsymbol{x}^* + \boldsymbol{q}(t). \tag{2.2}$$

前向误差为 $\Delta x(t) = x^* - \tilde{x}$, 它是数值解与精确解的距离, 也就是全局误差.

全局误差二范数为

$$\sqrt{\int_{t_0}^{t_m} \Delta oldsymbol{x}^{\mathrm{T}} \Delta oldsymbol{x} \ dt}.$$

由定义不难推导出以下定理, 其证明略.

定理 2.3 设 $\delta(t)$ 是线性常微分方程 (1.1) 的残差, 并且满足 $\delta = \tilde{x}' - A(t) \cdot \tilde{x} - q(t)$, 则残差满足以下线性常微分方程:

$$\Delta x' = A(t) \cdot \Delta x - \delta. \tag{2.3}$$

2.3 例子

考虑以下线性常微分方程的初值问题:

$$x(t)' = x(t), \quad x(0) = 1.$$

显然, 我们很容易写出它的精确解 $x^*(t) = e^t$. 假设时间为 $t \in [0,2]$, 我们简单地将时间区间等分为 [0,1] 和 [1,2] 两部分, 可见每一步的步长为 h = 1. 我们可以通过精确解计算求出每个格点上的数值解 $x_0 = 1$, $x_1 = e$, $x_2 = e^2$ 以及格点导数的数值解 $x_0' = 1$, $x_1' = e$, $x_2' = e^2$.

此时, 我们使用 Hermite 三次样条曲线进行插值, 得到结果如下:

$$\widetilde{x}(t) = \begin{cases} 1 + t + (2e - 5)t^2 + (3 - e)t^3, & t \in [0, 1], \\ e \cdot (t + (2e - 5)(t - 1)^2 + (3 - e)(t - 1)^3), & t \in [1, 2], \end{cases}$$

进一步, 我们写出前向误差 $\Delta x(t) = x^*(t) - \tilde{x}(t)$.

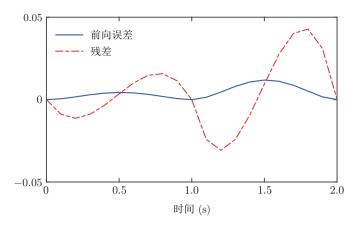


图 1 (网络版彩图) 举例说明的误差关系图

根据残差的定义公式 (2.1), 我们很容易计算出残差如下:

$$\delta(t) = \begin{cases} ((-3+e) \cdot t - 4e + 11) \cdot (t-1) \cdot t, & t \in [0,1], \\ e \cdot ((-3+e) \cdot t - 5e + 14) \cdot (t-1) \cdot (t-2), & t \in [1,2]. \end{cases}$$

综上所述, 利用 MATLAB 绘制误差关系图, 如图 1 所示, 不难发现, 在格点上, $\delta(0) = 0$, $\delta(1) = 0$, $\delta(2) = 0$, 这意味着采用 Hermite 三次样条曲线的插值法在 3 个格点都是无误差的, 但是在除了格点以外的地方都是有残差存在的.

3 线性常微分方程的误差估计

通常简化系统为线性时变的系统,它的状态矩阵 A(t) 导致了线性常微分方程很难精确地进行求解,因此需要假设状态矩阵 A(t) 随时间的变化不太过于剧烈,以保证系统误差估计效果明显.

定理 3.1 (Taylor 级数展开) 假设 $n \times n$ 矩阵 A(t) 在 t_0 的邻域范围内满足 $A(t) \in C^3$, 令 $A_0 = A(t_0)$, $A_1 = \frac{dA}{dt}|_{t=t_0}$, $A_2 = \frac{1}{2!} \cdot \frac{d^2A}{dt^2}|_{t=t_0}$, R(t) 为高阶余项, 那么有

$$\mathbf{A}(t) = \mathbf{A}_0 + \mathbf{A}_1 \cdot (t - t_0) + \mathbf{A}_2 (t - t_0)^2 + \mathbf{R}(t). \tag{3.1}$$

对于具有精确解的时不变系统, 无论选择何种数值解, 都可以进行精确的误差补偿; 而对于时变系统, 精确解的计算比较困难, 我们需要给出一个前向误差的上界.

定理 3.2 形如 (2.3) 的线性常微分方程的初值问题,设 $n \times n$ 常数矩阵 \mathbf{A}_0 为矩阵 $\mathbf{A}(t)$ 在 $t = t_0$ 时刻进行 Taylor 级数展开的线性部分,其中 $\{\lambda_i = a_i + b_i \cdot \mathbf{j}, i = 1, 2, \dots, n\}$ 为 \mathbf{A}_0 的特征根,按照实部由大到小排列, \mathbf{j} 为虚数单位,则一定存在一个可逆矩阵 \mathbf{P} ,满足 $\mathbf{A}_0 = \mathbf{P} \Sigma \mathbf{P}^{-1}$,其中 $\Sigma = \operatorname{diag}(a_1, a_2, \dots, a_n)$ 是对角矩阵.同时,假设

$$r_{\max} = \max_{i,k,t} \{(\|\boldsymbol{P}^{-1}(\boldsymbol{A}(t) - \boldsymbol{A}_0)\boldsymbol{P}\|_{\infty})_{i,k}\}, \quad \delta_{\max}(t) = \max_{t_0 \leqslant s \leqslant t} \|\boldsymbol{\delta}(s)_i\|_{\infty},$$

其中 $t \in [t_0, t_m]$, 则对于前向误差一定存在一个上界如下:

$$\|\Delta \boldsymbol{x}\|_{\infty} \leqslant \|\boldsymbol{P}\|_{\infty} \cdot \frac{\delta_{\max}}{a_1 + n \cdot r_{\max}} [e^{(a_1 + n \cdot r_{\max})(t - t_0)} - 1]. \tag{3.2}$$

证明 对线性常微分方程的初值问题 (2.3) 中的 A(t) 在初始时刻进行 Taylor 展开, 可以得到以下等式:

$$\Delta x' = A_0 \cdot \Delta x + (A(t) - A_0) \cdot \Delta x - \delta$$
$$= P \Sigma P^{-1} \cdot \Delta x + (A(t) - A_0) \cdot \Delta x - \delta.$$

对等式两边同时乘以 P^{-1} 矩阵进行变形, 然后令 n 维列向量 $\mathbf{y} = (y_1(t), y_2(t), \dots, y_n(t))^{\mathrm{T}} = \mathbf{P}^{-1} \Delta \mathbf{x}$, 那么一定有 $\mathbf{y}' = \mathbf{P}^{-1} \Delta \mathbf{x}'$. 对于初值问题, 容易知道 $\Delta \mathbf{x}(t_0) = \mathbf{0}$, 这意味着 $\mathbf{y}(t_0) = \mathbf{0}$. 利用上述关系 对等式进行约简, 得到

$$y' = \Sigma y + [P^{-1}(A(t) - A_0)P] \cdot y + (-P^{-1}\delta(t)).$$

对任意的正整数 i ($1 \le i \le n$) 不论 $y_i(t)$ 是何值, 根据上述等式可知, 它一定满足

$$y'_{i} = (a_{i} + b_{i} \cdot \mathbf{j})y_{i} + f_{i}(t), \quad y_{i}(0) = 0,$$

其中 $f_i(t)$ 是非线性部分. 进而, 很容易根据公式可以解出 y_i 如下:

$$y_i(t) = e^{(a_i + b_i \cdot \mathbf{j})t} \int_{t_0}^t f_i(\tau) e^{-(a_i + b_i \cdot \mathbf{j})\tau} d\tau.$$

进一步对上式进行缩放后有

$$|y_i(t)| \le e^{a_i t} \int_{t_0}^t |f_i(\tau)| e^{-a_i \tau} d\tau \le e^{a_1 t} \int_{t_0}^t |f_i(\tau)| e^{-a_1 \tau} d\tau.$$

假设 $z(t) = \max_{1 \leq i \leq n} |y_i(t)|$, 那么 $z(t) = ||\boldsymbol{y}||_{\infty}$. 又因为 z(t) 始终都是列向量 $|\boldsymbol{y}|$ 中的一个元素, 所以有 $|f_i(t)| \leq n \cdot r_{\max} z(t) + \delta_{\max}$, 因此,

$$z(t) \leq e^{a_1 t} \int_{t_0}^t (n \cdot r_{\text{max}} z(t) + \delta_{\text{max}}) e^{-a_i \tau} d\tau.$$

假设一个一维函数向量 $\phi(t)$, 其中 $\phi(t_0) = 0$, 满足

$$\phi(t) = e^{a_1 t} \int_{t_0}^t (n \cdot r_{\text{max}} \phi(t) + \delta_{\text{max}}) e^{-a_i \tau} d\tau,$$

对 $\phi(t)$ 进行求导, 有

$$\phi'(t) = (a_1 + n \cdot r_{\text{max}})\phi(t) + \delta_{\text{max}},$$

那么,很容易推导出

$$\phi(t) = \frac{\delta_{\text{max}}}{a_1 + n \cdot r_{\text{max}}} (e^{(a_1 + n \cdot r_{\text{max}})(t - t_0)} - 1).$$

最后, 由于 $z(t) \leq \phi(t)$, $\|\Delta x\|_{\infty} \leq \|P\|_{\infty} \cdot \|y\|_{\infty}$, 我们可以证明得到 (3.2) 的前向误差上界关系. 口对于线性常微分方程的初值问题而言, 误差估计的上界将随时间呈指数增长, 其增长速度主要取决于指数 $\lambda_1 + n \cdot r_{\max}$ 及它的残差 δ_{\max} . 如果动力系统是稳定的, 也就是 $\lambda_1 + n \cdot r_{\max} \leq 0$, 那么只要时间足够长, 系统误差上界终将会趋于一个常数. 相反地, 如果动力系统是不稳定的, 也就是 $\lambda_1 + n \cdot r_{\max} > 0$, 那么系统的误差上界将一直随时间急剧增大. 然而, 这种现象并不表明我们的误差估

计方法是不可靠的, 而是说明了我们的估计方法是非常恰当的, 下面将结合第 2.3 小节的例子来解释 这种情形.

例子中, 我们已知该常微分方程的精确解为 $x^* = e^t$, 如果在数值解中存在一个误差 ε , 满足

$$\widetilde{x}(t) = (1+\varepsilon)e^t - \varepsilon,$$

我们可以很容易计算出残差 $\delta(t) = \varepsilon$ 和前向误差 $\Delta x = x^* - \tilde{x} = \varepsilon(1 - e^t)$. 很显然, 在该例子中, 前向误差不可避免地是呈指数增长的. 利用定理 3.2 进行误差估计, 其中 $\lambda_1 = 1$, $r_{\max} = 0$, 我们的估计上界是 $\varepsilon(e^t - 1)$, 它同样也是呈指数增长的, 而且增长的速度与前向误差界的 $\|\Delta x\|_{\infty}$ 的增长速度相同, 这也意味着我们的误差估计方法十分有效, 而且很难再做较大的改进了.

Nedialkov 等 ^[25] 的区间求解方法需要采用高阶的 Taylor 级数展开对微分方程进行预处理, 然后才能够得到网格点所在的一个时间区间上的解区间, 该网格点上的精确解属于该解区间. 由于没有将数值解插值为近似解, 因而区间求解方法只能给出网格点所在区间上的解的误差估计区间, 无法给出全局的误差估计. 根据定理 2.2, 由于 Hermit 插值方法带来的余项误差造成本文估计方法的估计误差范围大于区间求解方法, 但这是由插值方法和步长所决定的, 不是误差估计方法本身的问题, 改进插值方法和步长能够有效地减小本文的误差估计上界. 同时, 本文的误差估计方法由于插值了整个区间的数值解, 不仅能够针对初值问题的 ODE 系统的误差估计, 还有望能够推广到边界值问题的 ODE 系统的误差估计.

注 3.1 本文的误差估计方法虽然形式上与 (1.2) 相似, 都为指数函数, 但本文的指数取决于最大的特征值实部; 而文献 [23] 的指数为 Lipschitz 常数, 主要取决于最大特征值的模. 当系统稳定时, 前向误差将逐渐趋于稳定, 最大的特征值实部为负数, 使得本文的误差估计上界也趋于稳定, 而 (1.2) 的误差估计上界却是不稳定的.

4 全局误差优化求解方法

为了减小线性常微分方程初值问题在数值计算中的全局误差,在固定网格点的条件下,我们希望解的全局误差范数最小.但是由于前向误差的不可解性,我们需要进一步将问题进行转化,由误差估计方法可知,前向误差的上界与残差成不等式关系 (3.2),也就是说,减小残差就能够降低前向误差的上界,从而实现减小全局误差.因此,我们将全局误差的优化问题转化为求解残差二范数的最小值的优化问题.

为了便于进行插值和计算, 我们将残差分为 m 段, 并对二范数进行平方处理, 定义目标方程如下:

$$Obj = \min \sum_{i=0}^{m-1} \int_{t_i}^{t_{i+1}} \boldsymbol{\delta}_i(t)^{\mathrm{T}} \cdot \boldsymbol{\delta}_i(t) dt,$$

$$(4.1)$$

其中 $\delta_i(t)$ 是第 i 个时间段上的残差函数.

采用 $\tau = (t - t_i)/h_i$ 将目标方程归一化处理为

Obj = min
$$\sum_{i=0}^{m-1} \frac{1}{h_i} \cdot \int_0^1 \boldsymbol{\delta}_i(\tau)^{\mathrm{T}} \cdot \boldsymbol{\delta}_i(\tau) d\tau$$
. (4.2)

通过定理 2.1, 我们可以利用 x_i 、 x_i' 、 x_{i+1} 和 x_{i+1}' 构造出 $\delta_i(t)$ 的三次样条曲线, 其中 x_i 、 x_i' 、 x_{i+1} 和

 x'_{i+1} 为未知的常数列向量. 设矩阵

$$a_i = \frac{\alpha'_0}{h_i} \mathbf{E} - \alpha_0 \mathbf{A} (\tau \cdot h_i + t_i),$$

$$b_i = \alpha'_1 \mathbf{E} - h_i \alpha_1 \mathbf{A} (\tau \cdot h_i + t_i),$$

$$c_i = \frac{\beta'_0}{h_i} \mathbf{E} - \beta_0 \mathbf{A} (\tau \cdot h_i + t_i),$$

$$d_i = \beta'_1 \mathbf{E} - h_i \beta_1 \mathbf{A} (\tau \cdot h_i + t_i),$$

其中 E 为 $n \times n$ 的单位矩阵. 结合线性常微分方程, 我们能够推导出

$$m{\delta}_i = m{a}_i m{x}_i + m{b}_i m{x}_i' + m{c}_i m{x}_{i+1} + m{d}_i m{x}_{i+1}' - m{q}.$$

进一步, 对于每一段残差, 假设 $I_i = \frac{1}{h_i} \cdot \int_0^1 \boldsymbol{\delta}_i(\tau)^{\mathrm{T}} \cdot \boldsymbol{\delta}_i(\tau) \ d\tau$, 那么有

$$I_{i} = \boldsymbol{y}_{i}^{\mathrm{T}} \boldsymbol{F}_{i} \boldsymbol{y}_{i} - 2\boldsymbol{B}_{i}^{\mathrm{T}} \boldsymbol{y}_{i} + \int_{0}^{1} \boldsymbol{q}^{\mathrm{T}} \boldsymbol{q} d\tau, \tag{4.3}$$

其中

$$egin{split} oldsymbol{F}_i = rac{1}{h_i} \cdot \int_0^1 egin{pmatrix} oldsymbol{a}_i^{\mathrm{T}} oldsymbol{a}_i & oldsymbol{a}_i^{\mathrm{T}} oldsymbol{b}_i & oldsymbol{a}_i^{\mathrm{T}} oldsymbol{c}_i & oldsymbol{a}_i^{\mathrm{T}} oldsymbol{c}_i & oldsymbol{a}_i^{\mathrm{T}} oldsymbol{c}_i & oldsymbol{a}_i^{\mathrm{T}} oldsymbol{c}_i & oldsymbol{b}_i^{\mathrm{T}} oldsymbol{c}_i & oldsymbol{a}_i^{\mathrm{T}} oldsymbol{c}_i & oldsymbol{c}_i^{\mathrm{T}} oldsymbol{c}_i^{\mathrm{T}} oldsymbol{c}_i & oldsymbol{c}_i^{\mathrm{T}} oldsymbol{c}_i^{\mathrm$$

是一个 $4n \times 4n$ 的矩阵,

$$oldsymbol{B}_i = -rac{1}{h_i} \cdot \int_0^1 oldsymbol{q}^{
m T} \cdot (oldsymbol{a}_i, oldsymbol{b}_i, oldsymbol{c}_i, oldsymbol{d}_i) d au$$

和

$$m{y}_i = (m{x}_i, m{x}_i', m{x}_{i+1}, m{x}_{i+1}')^{ ext{T}}$$

都是 4n 维列向量.

由于我们将优化求解问题的时间区间分解为 m 段相互独立的, 因此, 我们可以更进一步采用并行计算分别对 F_i 和 B_i 进行求解, 提高计算效率.

又因为 $\sum_{i=0}^{m-1} \frac{1}{h_i} \cdot \int_0^1 {\bf q}^{\rm T} {\bf q} dt \ge 0$ 为一个常数, 因此, 移除常数项后, 目标方程可以被改写为如下二次型的形式:

$$Obj = \min \mathbf{y}^{\mathrm{T}} \mathbf{F} \mathbf{y} - 2\mathbf{B}^{\mathrm{T}} \mathbf{y}, \tag{4.4}$$

其中 $\mathbf{y} = \{x_0, x_0', \dots, x_m, x_m'\}$, \mathbf{F} 为 $2n(m+1) \times 2n(m+1)$ 的大规模稀疏对称正定矩阵, \mathbf{B} 为 2n(m+1) 维的列向量, 结构如图 2 所示, 图中 "+" 表示两个相邻的矩阵块相加.

满秩的情形下,该二次型一定存在唯一解,根据文献 [33] 可知, 残差的最小值问题可以被转化为求解以下线性方程的数值解:

$$Fy = B. (4.5)$$

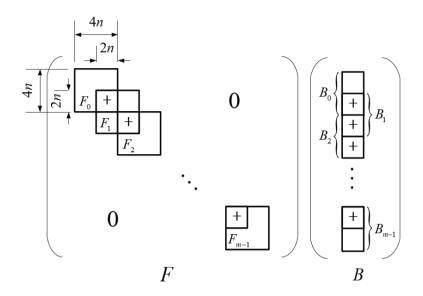


图 2 矩阵 F 和列向量 B 的结构示意图

众所周知, 利用共轭梯度法可以有效地求解这类线性系统. 特别地, 如果先通过其他求解器得到近似的数值解,则可以用这个数值解作为共轭梯度法的初始点来迭代求解.

在此必须指出, 我们的优化求解方法对于线性常微分方程的初值问题和边值问题都同样适用. 唯一的区别是, 在初值问题中, F 的前 n 行和前 n 列将在代入初值条件后被删除; 在边界值问题中, 与边界值相对应的有 n 行和 n 列将被删除. 同样的处理也发生在 B 中, 因此, (4.4) 的未知数减少了 n 个.

注 4.1 本文优化求解方法的主要贡献在于, 将全局误差最小问题转化为残差二范数最小问题, 通过定义插值空间及一系列的形式转化, 将残差二范数最小值问题转化为二次型等价的线性方程组求解问题, 最终将微分代数方程的求解问题转化为由 (4.3) 直接构造特殊结构的线性方程组, 并可采用共轭梯度法高效求解.

5 实验对比

5.1 误差估计例子

针对线性时变的常微分方程系统的初值问题,由于通常无法直接求出它的精确解,为了展示误差估计方法的实验效果,我们可以利用一个已知的精确解去构造一个线性时变的常微分方程系统,进一步通过设定状态矩阵的特征值为负将系统构造为一个稳定的系统,具体的构造如下:已知

$$\mathbf{A}(t) = \begin{pmatrix} -6 + 0.2 \cdot \sin(2\pi t) & 0\\ 1 & -5 - 0.1 \cdot \sin(\pi t) \end{pmatrix},$$

以及

$$x = \begin{pmatrix} \cos(\pi t) - 1 \\ \sin(\pi t) \end{pmatrix},$$

由于存在 $\mathbf{q} = \mathbf{x}' - \mathbf{A}(t) \cdot \mathbf{x}$ 的关系, 我们可以进一步计算出 \mathbf{q} , 假设时间 $t \in [0,2]$, 很容易可以得出初始条件

$$\mathbf{x}(0) = (0,0)^{\mathrm{T}}.$$

首先利用 MATLAB 提供的求解器 ODE45 对该系统进行数值求解, 计算出 x 和 x'; 然后使用 Hermit 三次样条曲线对数值解进行插值得出近似解, 并由 (2.1) 和前向误差的定义推导计算出残差和前向误差数值并绘制如图 3 所示的残差和前向误差, 其中, 本实验系统为二维系统, 因而对应每一个维度分别有一条与之对应的残差线和前向误差线, 在后续实验例子中同理; 最后, 根据定理 3.2, 我们对该系统进行误差估计并绘制如图 3 所示的误差估计上界, 误差估计上界唯一, 且能够很好地将系统的任意一个前向误差都控制在一个数量级范围内, 且随着时间的增加而趋于稳定, 这与该系统的稳定性保持一致.

5.2 优化求解例子

为了更好地对比现有的求解方法, 首先采用 MATLAB 的 ODE45 求解器对线性常微分方程的初值问题进行求解, 再在相同的网格下, 采用我们的优化求解方法进行对比实验. 进一步, 为了更好地说明问题, 我们可以将 \boldsymbol{A} 设定为一个常数矩阵, \boldsymbol{q} 设定为一个常数列向量, 那么就能准确地得到该系统的精确解 \boldsymbol{x}^* , 并计算出准确的前向误差.

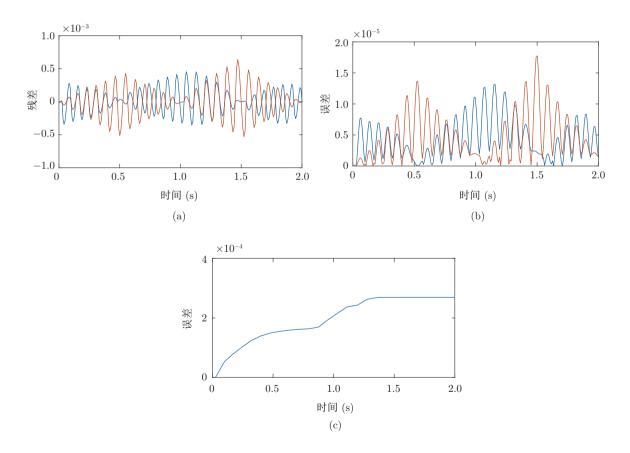


图 3 (网络版彩图) 线性时变的常微分方程系统的误差估计结果. (a) 残差; (b) 前向误差; (c) 误差估计上界. 图中不同的颜色代表不同的维度

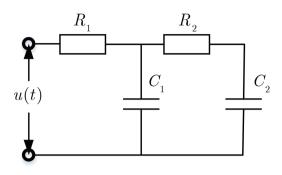


图 4 RC 并联滤波电路

RC (resistor-capacitance circuit) 滤波电路在电力电子和控制领域应用得相当广泛, 它可以通过阻塞某些频率信号和传递其他频率信号进行滤波. 如图 4 所示的 RC 并联滤波电路, 设状态变量 $\boldsymbol{x} = [x_1,x_2]^{\mathrm{T}}$ 为对应每个电容上的电压, 在时间 $t \in [0,2]$ 上, 我们对该系统在输入信号作用下随时间的响应进行研究, 可以将其看作为一个初值问题. 假定电阻阻值分别为 $R_1 = R_2 = 10 \,\mathrm{k}\Omega$, 电容大小分别为 $C_1 = C_2 = 100 \,\mathrm{\mu F}$. 初始时刻, $\boldsymbol{x}(0) = [0,0]^{\mathrm{T}}$, 我们给系统输入端加入 $u(t) = 2 \cdot \sin(2t)$ 的电压, 并建立数学模型如下:

$$m{x}' = \begin{pmatrix} -2 & 1 \\ 1 & -1 \end{pmatrix} m{x} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(t).$$

显然, 该线性常微分方程系统的精确解为

$$\boldsymbol{x}^* = e^{\boldsymbol{A}t} \cdot \int_0^t e^{-\boldsymbol{A}s} \boldsymbol{q}(s) ds.$$

在相对误差为 10^{-5} 、采用 ODE45 进行求解时, 求解器将时间区间划分为不均匀的 13 个网格点, 在相同的网格点上, 我们使用优化求解方法进行对比实验. 其中, 优化求解方法构造的矩阵 F 为 50×50 的矩阵, 它的条件数为 3.949×10^3 . 优化求解方法与 ODE45 的残差二范数大小几乎相同, 分别为 0.4097×10^{-3} 和 0.4275×10^{-3} ,残差变化如图 5 所示, 两者也很相近. 然而, 它们的前向误差却相差十分明显, 优化求解方法的前向误差波动幅度明显较小. 图中对比更加明显的是全局误差二范数, 优化求解方法的全局误差二范数在终止时刻为 0.1264×10^{-4} , 明显小于 ODE45 的 0.2336×10^{-4} .

5.3 主动式汽车悬架系统

为了车厢内驾驶人员和乘客乘坐的舒适感,在汽车通过一个不平整的路面时,我们总是希望汽车能够足够地平稳,通常在设计过程中采用悬架系统来解决这一问题.目前最普遍的悬架系统为主动式汽车悬架系统,主要通过电脑辅助实时调整阻尼器的阻值大小再结合固定的减震器来实现.在设计之初,研发人员通常需要对动力系统进行数学建模分析,以便能够精确地模拟出现实路况中的汽车性能.然而,该数学模型的数值解如果不够准确、快速,反馈给控制系统可能会带来灾难性的后果.为了能够实时准确地实现控制和预警,我们可以对系统进行优化求解,以及针对优化解给出一个前向误差界估计,如果前向误差的界太大,则反馈研发人员该计算结果不可用.

如图 6 所示, 以简单的主动式汽车悬架系统为例. 在垂直方向上, 假设单个轮胎重量 $m_1=100$ kg, 轮胎弹性系数 $k_1=2000$ N/m, 部分车身重量 $m_2=500$ kg, 减震弹簧弹性系数 $k_2=1500$ N/m. 假定该汽车在 $t\in[0,0.5]$ 内通过一个高度为 0.2 m 的障碍物. 为了保证汽车车身的平稳, 当主动式阻尼的阻尼系数 $\eta_2=200+200\cdot(t-0.25)^2$ N·m/s² 时, 研究汽车行驶通过该坡期间的状态.

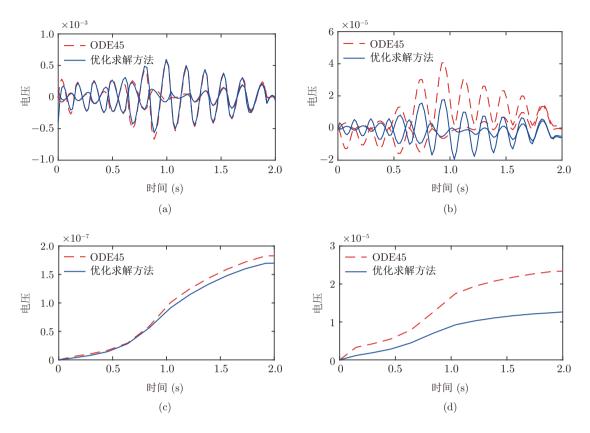


图 5 (网络版彩图) ODE45 (虚线) 和优化求解方法 (实线) 在例子中的结果对比. (a) 残差; (b) 向前误差; (c) 残差二范数; (d) 整体误差二范数

该数学模型为一个线性 ODE 系统,

$$x' = A(t) \cdot x + q,$$

其中 $\mathbf{x} = (x_1, x_1', x_2, x_2')^{\mathrm{T}}$ 分别为轮胎的位移和速度以及汽车车身的位移和速度, 初始时刻 $\mathbf{x}(\mathbf{0}) = (\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0})^{\mathrm{T}}$. 为了能够获取精确解, 以便于验证, 假设车身平稳的要求为

$$\boldsymbol{x}^* = (0.2 \cdot [(t - 0.25)^2 - 0.25^2], 0.4 \cdot t, 0.02 \cdot [(t - 0.25)^2 - 0.25^2], 0.04 \cdot t)^{\mathrm{T}},$$

则可以根据 $q = (x^*)' - A(t) \cdot x^*$ 求出该障碍物的相关输入信息. 进一步, 得出该系统的状态矩阵 A 为

$$\mathbf{A}(t) = \begin{pmatrix} 0 & 1 & 0 & 0\\ 35 & 2 + 2 \cdot (t - 0.25)^2 & -15 & -2 - 2 \cdot (t - 0.25)^2\\ 0 & 0 & 0 & 1\\ 3 & \frac{2}{5} + \frac{2}{5} \cdot (t - 0.25)^2 & -3 & -\frac{2}{5} - \frac{2}{5} \cdot (t - 0.25)^2 \end{pmatrix}.$$

首先, 我们分别采用 ODE45 方法和优化求解方法对该线性 ODE 系统模型进行求解, 实验结果如图 7 所示, 虚线对应 ODE45 方法, 实线对应优化求解方法. 其中相同网格点条件下的优化求解方法的数值解的前向误差为 10⁻¹¹ 数量级, 远小于 ODE45 方法的 10⁻⁶ 数量级; 同时, 就残差而言, 优化求解方法也是远小于 ODE45 方法 5 个数量级左右. 由于该汽车悬架系统模型是一个时变不稳定系统. 因

此,数值解的前向误差随时间呈指数增长,但优化求解方法的增长率远低于 ODE45 方法. 故而,采用优化求解方法进行计算更加准确,离线模式更加有利于仿真的实时性.

然后,我们分别对 ODE45 方法和优化求解方法的数值解的误差进行估计,并通过已知的前向误差进行验证,对比实验结果如图 8 所示,虚线对应各个维度的前向误差,实线对应误差估计上界.系统不稳定导致的前向误差指数增长与误差估计基本保持一致,也就是说,对于不稳定系统,如果我们需要足够的误差估计精度,时间应该足够短;由于数值解准确性的不同,误差估计的界限会有很大的不同,数值解越准确,误差估计上界越小.同时,误差估计方法对两种求解方法数值解给出的误差估计上界都是在同一个数量级的小范围内对前向误差进行了有效的界定.

5.4 性能对比实验

本节所有的性能对比实验都是在 Intel(R) Core(TM) I5-7400 CPU @3.00GHz 的计算机上进行的. 其中, VNODE-LP 区间算法的 C++ 安装包由 http://www.cas.mcmaster.ca/~nedialk/vnodelp/ 提供,在 Ubuntu 操作系统下运行; ODE45 由 MATLAB 软件提供. 误差估计方法也是在 MATLAB 软件上根据定理 3.2 的结论编写算法程序进行实现的. 优化求解算法由两部分构成: 一是由 (4.3) 推导来构造特殊结构矩阵的符号计算的预处理部分 (off-line), 具体为在 Maple 软件上编写算法程序实现符号矩阵的构造; 二是由 (4.5) 推导而来的赋值和数值求解部分构成 (on-line), 具体为在 Maple 软件中利用subs 赋值指令生成数值矩阵, 然后在 MATLAB 软件上求解线性方程数值解.

5.4.1 误差估计性能对比

为了更好地展示我们的误差估计方法的性能, 我们将本文误差估计方法与现有的 VNODE-LP 区间算法的误差估计进行对比实验. 在 $t \in [0,0.1]$ 的区间上, 随机生成特征值为负、系数范围在 [-10,10]

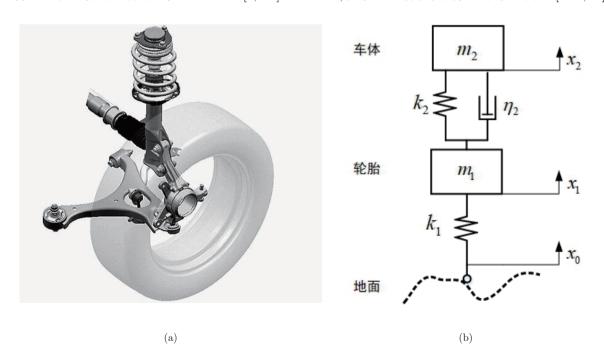


图 6 汽车悬架系统. (a) 汽车悬架系统结构图 (图片来自 https://baike.baidu.com/item/麦弗逊式独立悬架/10950807?fr=aladdin); (b) 汽车悬架系统受力示意图

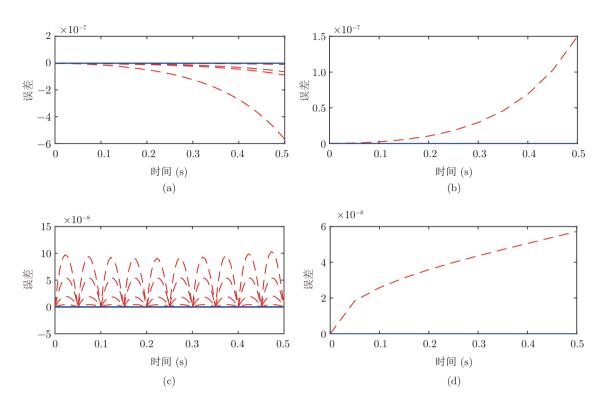


图 7 (网络版彩图) 汽车悬架系统求解实验结果对比. (a) 前向误差; (b) 整体误差; (c) 残差; (d) 残差二范数. 图中不同的颜色代表不同的维度

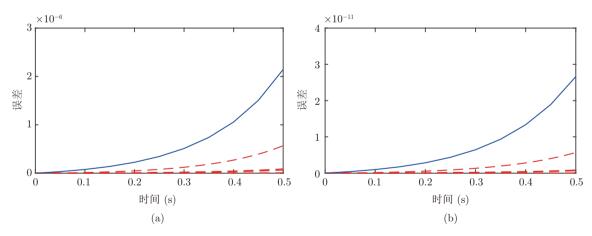


图 8 (网络版彩图) 汽车悬架系统误差估计实验结果对比. (a) ODE45 误差估计上界; (b) 优化求解方法误差估计上界. 图中不同的颜色代表不同的维度

的常系数矩阵 A, q 为零向量. 设定 VNODE-LP 区间算法的 Order 为 5, 其中 VNODE-LP 区间算法 给出的数值解区间的一半为误差估计的精度, 我们的误差估计方法对 ODE45 计算的结果进行误差估计. 分别对不同维度的系统进行 10 次实验后求均值, 由于维度过大会引起符号计算求解精确解的计算量太大, 因而本实验的维度未能选取太大, 但也已经能够展示出结果的差异. 实验结果如表 1 所示, 可见, 我们的误差估计方法在误差估计的精度上优于 VNODE-LP 区间算法近一个数量级, 且与 δ_{max} 呈

农 I VNODE-LF 相庆左旧月月本住肥利几						
	VNODE-LP (Order 5) 误差估计方		计方法			
	估计精度	估计精度	$\delta_{ m max}$			
3×3	2.10987E-7	0.862E - 8	4.7888E - 7			
4×4	$1.64408E{-7}$	$3.426E{-8}$	$1.1935E{-6}$			
5×5	$8.48887E{-7}$	2.779E - 8	1.1160E - 6			
6×6	$1.13325E{-6}$	1.317E - 7	$4.1681E{-6}$			
7×7	$1.51515E{-6}$	$1.211E{-7}$	$3.7651E{-6}$			
10 × 10	$3.84512E{-6}$	$3.100E{-7}$	7.4032E - 6			

表 1 VNODE-LP 和误差估计方法性能对比

正相关,同时,随着系统维度的增大也会造成误差估计的精度降低.

5.4.2 数值求解性能对比

为了更好地展示优化求解方法的性能、我们与 MATLAB 的 ODE45 方法进行数值求解对比实验. 首先, 需要随机构造 $n \times n$ 状态矩阵

$$\mathbf{A}(t) = \operatorname{diag}(-1) + \mathbf{m}_1 \cdot \sin(\mathbf{m}_2 \cdot t) + \mathbf{m}_3 \cdot (t^{\mathbf{m}_4} - 1),$$

以及 n 维列向量 $q(t) = m_5 \cdot \sin(m_6 \cdot t)$, 其中 $m_1 \cdot m_3$ 和 m_5 为随机稀疏的 0-1 矩阵

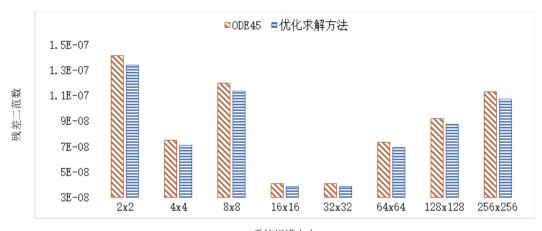
$$\mathrm{randsrc}\bigg(n,n,\left\lceil 1,0;\frac{1}{n},1-\frac{1}{n}\right]\bigg),$$

 m_2 和 m_4 为随机稀疏的 0-1-2 矩阵

$$\operatorname{randsrc}\bigg(n,n,\left[2,1,0;\frac{1}{n^2},\frac{1}{n^2},1-\frac{2}{n^2}\right]\bigg),$$

 m_6 为随机整数向量 randi([-2,2],n,1). 然后, 假定时间 $t \in [0,0.1]$, 设定 t=0 时刻的初值都为 1. 最 后, 利用构造的系统分别用 ODE45 和优化求解方法在不同维度上进行 10 次实验后求均值, 得到实验 结果如表 2 和图 9 所示.

表 2 数值求解方法性能对比表 2 数值求解方法性能对比								
	ODE45		优化求解方法					
大小	(MATLAB)	离线	在线					
		预处理 (Maple)	赋值 (Maple)	CGM	MATLAB			
2×2	0.746	4.571	0.119	0.028	0.022			
4×4	0.891	7.185	0.144	0.041	0.216			
8×8	1.077	13.521	0.212	0.077	0.003			
16×16	2.185	15.964	0.321	0.229	0.043			
32×32	73.806	18.500	0.665	0.728	0.234			
64×64	109.376	76.688	1.834	11.313	1.262			
128×128	158.329	208.719	7.344	59.264	10.005			
256×256	423.307	954.380	34.428	278.196	96.903			



系统规模大小

图 9 (网络版彩图) 残差二范数对比图

由于无法获取精确解, 我们可以通过计算残差, 然后用对比残差二范数的方式比较两种方法的计算精度. 如图 9 所示的残差二范数对比图, 优化求解方法的残差在各个维度上都小于 ODE45 的残差. 特别需要补充说明的是, 我们在同一维度每次实验优化求解方法的残差都是较优的, 由于数据太多且效果与第 5.2 和 5.3 小节的实验结果一致, 因此没有一一罗列.

表 2 的单位都是秒 (s), 其中数值求解方法分别采用了我们编写的共轭梯度法 (conjugate gradient method, CGM) 和 MATLAB 自带的解线性方程的方法. 分析表 2 所示的结果, 可见: (1) 纵向看, 随着维度的增加, 计算时间都呈现出类似指数型的增加; (2) 横向看, 同为对进行时间对比, ODE45 耗时远优于离线预处理部分的耗时, 但又大幅度高于执行赋值程序和 CGM 程序的总时间或者执行赋值和 MATLAB 的在线处理的总时间; (3) 相对而言, 未优化的 CGM 在求解效率上还是远低于 MATLAB 自带解线性方程的方法. 例如, 当 n=256 时, ODE45 用时为 423.307 s, 不足离线耗时的一半, 如果已经采用离线构造好符号矩阵, 则只需要在线进行赋值和数值求解; 其中执行赋值程序和 CGM 程序的总时间 (34.428 + 278.196) s 约为 ODE45 耗时的 3/4, 执行赋值程序和 MATLAB 自带求解程序的总时间 (34.428 + 96.903) s 约为 ODE45 耗时的 1/3, 造成这一现象的原因在于 F 矩阵的带状特殊结构以及我们未对 CGM 进行算法优化, 针对这一现象,我们将在下一步的工作中提出解决方案.

注 5.1 优化求解方法主要计算量都在构造特殊矩阵的符号求解过程,但对于结构相同、数值不同的线性常微分方程,例如,针对同一系统做多次求解计算,我们可以只离线进行一次预处理,之后就可以直接进行在线的赋值和数值求解,极大地提高了计算效率.

6 总结

为了回答线性常微分方程的初值问题中如何确定它的全局误差上界和提高它解的质量这两个问题,我们在定义残差的基础上分别提出了误差估计方法和优化求解方法.

经过理论推导和实验验证,误差估计方法能够较好地实现对数值解的前向误差上界的估计,其估计的上界主要取决于最大特征值的实部,通常小于 Lipschitz 常数. 当系统稳定时,误差估计上界趋于稳定,与第 5.1 小节结果一致;当系统不稳定时,误差估计上界呈指数增长,且增长速度与前向误差保持一致,与第 5.3 小节结果一致;从第 5.4.1 小节的性能对比实验结果可知,在相同条件下,误差估计方法的估计精度优于 VNODE-LP 方法.

对于优化求解方法,我们提出了一种基于共轭梯度的线性常微分方程最优解计算方法,仅通过求解构造的线性方程处理线性常微分方程问题,可以直接得到全局最优解.同时,构造的大规模稀疏矩阵 F 具有良好的结构特性,整个过程可利用并行计算方法进行求解,只需要对 4n×4n 大小的符号矩阵进行符号积分求解,然后分别代入网格点值和状态矩阵求解其数值矩阵.该优化求解方法能够求得数值解在插值空间上的最小残差解,与 MATLAB 自带的求解器进行大量对比实验表明,在网格相同的情形下,该方法整体都优于现有的数值求解器的方法.同时,通过实验 5.4.2 的计算性能对比结果可知,优化求解方法特别适用于先离线进行预处理,然后再进行在线赋值和数值计算的场景,此种组合计算方法效率达到最高.

不难发现,本文的优化求解方法不仅适用于线性常微分方程系统的初值问题,同时还适用于边界值问题.下一步工作将研究线性常微分方程边界值问题的误差估计,并基于误差估计上界给出相应的边界值问题优化求解方法.

致谢 感谢两位匿名审稿人对本文提出的宝贵建议.

参考文献 _

- 1 Platzer A, Quesel J D. European train control system: A case study in formal verification. In: Formal Methods and Software Engineering. Lecture Notes in Computer Science, vol. 5885. Berlin-Heidelberg: Springer, 2009, 246–265
- 2 Pnueli A, Sifakis J. Special issue on hybrid systems. Theoret Comput Sci, 1995, 138: 1-2
- 3 Johnson T T, Green J, Mitra S, et al. Satellite rendezvous and conjunction avoidance: Case studies in verification of non-linear hybrid systems. In: Formal Methods. Lecture Notes in Computer Science, vol. 7436. Berlin-Heidelberg: Springer, 2012, 252–266
- 4 Walter W. Ordinary Differential Equations. New York: Springer, 1998
- 5 Ascher U M, Mattheij R, Russell R. Numerical Solution of Boundary Value Problems for Ordinary Differential Equations. Philadelphia: SIAM, 1988
- 6 Shampine L F, Kierzenka J, Reichelt M W. Solving boundary value problems for ordinary differential equations in MATLAB with bvp4c. Https://classes.engineering.wustl.edu/che512/bvp_paper.pdf, 2000
- 7 Shampine L F, Reichelt M W. The MATLAB ODE suite. SIAM J Sci Comput, 1997, 18: 1–22
- 8 Shampine L F, Corless R M. Initial value problems for ODEs in problem solving environments. J Comput Appl Math, 2000, 125: 31–40
- 9 Shampine L F. Error estimation and control for ODEs. J Sci Comput, 2005, 25: 3-16
- 10 Shampine L F. Solving ODEs and DDEs with residual control. Appl Numer Math, 2005, 52: 113-127
- 11 Corless R M, Fillion N. A Graduate Introduction to Numerical Methods. New York: Springer, 2013
- 12 Higham D J. Robust defect control with Runge-Kutta schemes. SIAM J Numer Anal, 1989, 26: 1175-1183
- 13 Constantinescu E M. Generalizing global error estimation for ordinary differential equations by using coupled timestepping methods. J Comput Appl Math, 2018, 332: 140–158
- 14 Enright W H. Continuous numerical methods for ODEs with defect control. J Comput Appl Math, 2000, 125: 159–170
- 15 Higham D J. Runge-Kutta defect control using Hermite-Birkhoff interpolation. SIAM J Sci Stat Comput, 1991, 12: 991–999
- 16 Calvo M C, Higham D J, Montijano J I, et al. Stepsize selection for tolerance proportionality in explicit Runge-Kutta codes. Adv Comput Math, 1997, 7: 361–382
- 17 Shampine L F. Tolerance Proportionality in ODE Codes. Berlin-Heidelberg: Springer, 1989
- 18 Stetter H J. Tolerance proportionality in ODE-codes. In: Proceedings of the 2nd Conference on Numerical Treatment of Ordinary Differential Equations. Berlin: Humboldt University Press, 1980, 109–123
- 19 Dormand J R, Duckers R R, Prince P J. Global error estimation with Runge-Kutta methods. IMA J Numer Anal, 1984, 4: 169–184
- 20 Dormand J.R., Prince P.J. Global error estimation with Runge-Kutta methods II. IMA J. Numer Anal, 1985, 5: 481–497
- 21 Shampine L F, Watts H A. Global error estimates for ordinary differential equations. ACM Trans Math Software, 1976, 2: 172–186
- 22 Shampine L F, Watts H A. Algorithm 504: GERK: Global error estimation for ordinary differential equations. ACM Trans Math Software, 1976, 2: 200–203

- 23 Ascher U M, Petzold L R. Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations. Philadelphia: SIAM, 1998
- 24 Viswanath D. Global errors of numerical ODE solvers and Lyapunov's theory of stability. IMA J Numer Anal, 2001, 21: 387–406
- 25 Nedialkov N, Jackson K R, Corliss G F. Validated solutions of initial value problems for ordinary differential equations. Appl Math Comput, 1999, 105: 21–68
- 26 Frank R, Ueberhuber C W. Iterated defect correction for the efficient solution of stiff systems of ordinary differential equations. BIT, 1977, 17: 146–159
- 27 Stetter H J. The defect correction principle and discretization methods. Numer Math, 1978, 29: 425-443
- 28 Kreyszig E. Advanced Engineering Mathematics. New York: John Wiley & Sons, 2005
- 29 Verner J H. Differentiable interpolants for high-order Runge-Kutta methods. SIAM J Numer Anal, 1993, 30: 1446–1466
- 30 Enright W H, Hayes W B. Robust and reliable defect control for Runge-Kutta methods. ACM Trans Math Software, 2007, 33: 1
- 31 Enright W H, Muir P H. New interpolants for asymptotically correct defect control of BVODEs. Numer Algorithms, 2010, 53: 219–238
- 32 Endre S, David M. An Introduction to Numerical Analysis. Cambridge: Cambridge University Press, 2003
- 33 Shewchuk J R. An introduction to the conjugate gradient method without the agonizing pain. Https://www.cs.cmu.edu/~quake-papers/painless-conjugate-gradient.pdf

Global error estimation for linear ordinary differential equations and their numerical optimal solutions

Wengiang Yang & Wenyuan Wu

Abstract Solving initial value problems (IVPs) for linear ordinary differential equations (ODEs) plays an important role in many applications. There are various numerical methods and solvers to obtain approximate solutions typically represented by points. However, few works about estimation of global errors can be found in literature. In this paper, we first use Hermite cubic spline interpolation at mesh points to represent the solution, and then we define this residual obtained by substituting the interpolation solution back to ODEs. Then the global error between the exact solution and an approximate solution can be bounded by using the residual. Moreover, solving ODEs can be reduced to an optimization problem of the residual in certain solution space which can be solved by the conjugate gradient method by taking advantage of sparsity of the corresponding matrix. The examples in the paper show that our estimation works well for linear ODE models and the refinement can find solutions with smaller global errors than some popular methods in MATLAB without additional mesh points.

Keywords linear ordinary differential equation, global error, residual, refinement

 ${\rm MSC}(2010)\quad 34{\rm A}30,\, 34{\rm B}07,\, 34{\rm D}23,\, 34{\rm H}05$

doi: 10.1360/SSM-2019-0313