



吉林大学远程教育

本科生毕业论文（设计）

中文题目 在时间并行算法中学习粗网格
传播器

学生姓名: 小明 专业: 计算机科学技术

层次年级: 2020 本科 学号: 2111110101101

指导教师: _____ 职称: 教授

学习中心: 长春 成绩: _____

2024 年 3 月 14 日

摘 要:

时间并行算法中的一个重要类别是 Parareal 算法, 它广泛应用于解决演化方程. 为了实现有效的加速, 算法中选择合适的粗网格传播器至关重要. 在这项工作中, 我们探讨了学习型粗网格传播器的使用. 基于误差估计框架, 我们提出了一种构建具有稳定性和相容精度的粗网格传播器的方法. 此外, 我们为所得 Parareal 算法提供了初步的数学保证. 在各种设置上的数值实验, 例如线性扩散模型、Allen-Cahn 模型和粘性 Burgers 模型, 表明与一些传统且广泛使用的粗网格传播器相比, 学习型粗网格传播器可以显著提高并行效率.

关键词: 椭圆问题; 单步法; 收敛方法; 机器学习

目 录

第 1 章 绪论	4
1.1 研究背景和问题模型	4
1.2 本文贡献和文章架构	4
第 2 章 单步方法和并行算法	5
2.1 抛物方程的单步求解器	5
2.2 并行算法	7
第 3 章 总结	8
参考文献	9
致谢	11
附录	12
A 显式公式	12

第 1 章 绪论

1.1 研究背景和问题模型

本文致力于使用机器学习技术加速一类针对抛物问题的 Parareal 求解器. 设定一个固定的终止时间 $T > 0$. 考虑以下初始值问题, 对于 $u \in C((0, T]; D(\mathcal{A})) \cap C([0, T]; \mathcal{H})$:

$$\begin{cases} u'(t) + \mathcal{A}u(t) = f(t), & 0 < t < T, \\ u(0) = u^0, \end{cases} \quad (1.1)$$

其中 $u^0 \in \mathcal{H}$ 是初始数据, $f: [0, T] \rightarrow \mathcal{H}$ 是给定的强制项, 而 \mathcal{A} 是一个正定的、自伴的、线性算子, 拥有紧逆的算子, 定义在带有内积 (\cdot, \cdot) 的希尔伯特空间 \mathcal{H} 上 (带有诱导范数 $\|\cdot\|$), 其定义域 $D(\mathcal{A})$ 在 \mathcal{H} 中稠密. 这个模型在实际应用中有非常广泛的应用, 其有效的数值模拟具有极大的重要性, 并受到了大量关注 ([27]).

模型 (1.1) 的数值解通常采用时间离散化, 提出了各种时间步进格式; 具体内容参见专著 [27]. 这些格式通常逐步进行, 过程本质上是顺序的. 这是模型 (1.1) 数值解的主要瓶颈, 尤其是在大时间尺度上. 在开创性的工作 [21] 中, Lions、Maday 和 Turinici 提出了一种称为 Parareal 算法的方法, 用于以时间并行的方式求解演化模型. 该方法已成为在分布式计算在空间上饱和时利用高度并行系统的基础工具. 自从它的诞生以来, 这类方法已成功应用于众多不同的实际应用, 例如期权定价 [4, 25]、多尺度动力学 [2, 10]、随机 PDE [6] 和分数扩散 [20, 30].

Parareal 算法基于两个求解器, 一个是粗网格传播器 (coarse propagator, CP), 精度低但足够快以便顺序应用, 另一个是精度高的细网格传播器 (fine propagator, FP), 在时间上并行应用. 在每次迭代中, CP 在每个子域上顺序运行, 然后 FP 作为修正并行运行. 有关 CP 和 FP 如何交互的详细信息, 请参阅第 2.2 节. CP 通常基于标准数值求解器的一步或几步, 实际上, 它通常是使用较低分辨率的数值积分器, 例如更大的时间步长大小, 以有效地向前传播信息. 理想情况下, Parareal 算法可以以 CP 的效率达到 FP 的精度. 在现代多核计算机架构上实现所需的加速, CP 的选择至关重要. 尽管如此, CP 的最佳选择仍然极为不平凡. 更糟糕的是, 对于带有非平滑数据的模型 (1.1), 使用标准向后欧拉 CP 的 Parareal 算法的收敛性在某种程度上受限 [11, 22, 28, 32], 这本质上限制了可实现的效率, 参见下面 (2.4) 的精确陈述. 这个问题也出现在其他常用的 CP 上 [11, 表 1 和 2]. 这些观察自然引出了一个有趣的问题: 是否可以通过适当修改 CP 来提高 Parareal 算法的收敛速度.

1.2 本文贡献和文章架构

在这项工作中, 我们提出了一个创新且易于实现的框架, 用于设计针对问题 (1.1) 的 Parareal 算法中的 CP. 它基于误差估计框架, 并遵循使用学习来解决科学计算挑战的普遍范式 ([11, 14, 26, 32]). 具体来说, 基于在误差估计框架中推导出的分析收敛因子, 我们构造了一个学习型粗网格传播器 (learned coarse propagator, LCP), 以最小化收敛因子的上界.

这样, LCP 就是针对 FP 特别学习的, 这与常规选择中 CP 几乎对 FP 的选择无知大相径庭. 我们提供了一种构建拥有理想特性 (例如稳定性和准确性) 的 CP 的方法. 我们对该框架进行了初步分析, 表明 LCP 确实可以实现线性演化问题的预期目标. 我们展示了几个线性和非线性问题的数值实验, 包括线性扩散模型、Allen-Cahn 模型和粘性 Burgers 方程, 以说明这种方法的优越性. 数值结果表明, 与使用后向欧拉格式 (或其他流行的单步方法) 作为 CP 的传统选择相比, 使用学习技术始终能够提高性能.

这项工作大体上, 遵循了用于科学计算的学习数值算法的普遍范式, 它结合了数学驱动的、手工设计的一般算法结构与针对特定任务类别的数据驱动适应 ([5, 15, 16, 17, 18, 23]). 例如, Guo 等人提出了一种机器学习方法, 它能够自动学习求解常微分方程初值问题的有效解算器, 基于 Runge-Kutta 积分器, 并学习针对特定族群的微分方程的高阶积分器 ([16]). LCP 在设计新的并行算法中的应用也得到了探索 ([1, 18, 24, 31]). Yalla 和 Engquist 探索了使用神经网络近似高维谐振子和局部多尺度问题的相位图作为 CP([31]). Agboh 等人使用前馈深度神经网络作为 CP 来积分 ODE, 并观察到性能的提升 ([1]). Nguyen 和 Tsai 使用监督学习来提高波传播的效率 ([24]). Ibrahim 等人提出使用物理信息神经网络 (PINN) 作为 CP, 并展示了其在 Black-Scholes 方程上相比数值 CP 的加速性能 ([18]). 本文继续沿着这一活跃的研究路线探索使用学习概念增强并行算法 ([1, 18, 24, 31]). 但与众不同的是, 它显式地将分析洞察融入学习过程中, 即稳定性和相容性, 因此本文提出的 LCP 具有严格的数学保证 ([19]). 更具体地说, 我们的工作可以概括为:

- (i) 设计算法学习 Parareal 算法中的学习型粗网格传播器.
- (ii) 提供理论保证, 学习型粗网格传播器关于网格比是鲁棒的.
- (iii) 在三种抛物问题中的数值案例中, 说明学习型粗网格传播器大大加速了并行效率.

第 2 章 单步方法和并行算法

首先, 我们描述了用于解决问题 (1.1) 的单步法的抽象框架, 并简要概述了并行算法. 对于这些主题的深入处理, 我们建议感兴趣的读者参考专著 [27] 和综述 [12].

2.1 抛物方程的单步求解器

为了在时间上离散化问题 (1.1), 我们将时间区间 $(0, T)$ 分为 N 个等距子区间, 每个子区间长度为 $\Delta t = T/N$, 并设格点为 $t_n = n\Delta t$, 其中 $n = 0, 1, \dots, N$. 然后, 单步求解器通过

$$u^{n+1} = r(\Delta t \mathcal{A})u^n + \Delta t \sum_{i=1}^m p_i(\Delta t \mathcal{A})f(t_n + c_i \Delta t), \quad \text{对于 } 0 \leq n \leq N-1. \quad (2.1)$$

来近似解 $u(t_{n+1})$. 这里 $r(\cdot)$ 和序列 $\{p_i(\cdot)\}_{i=1}^m$ 是有理函数, c_i 是 $[0, 1]$ 中的不同实数. 下面我们对格式 (2.1) 提出以下条件:

(P1): $|r(\lambda)| < 1$ 和 $|p_i(\lambda)| \leq c$, 对所有 $i = 1, \dots, m$, 对于任何 Δt 和任何 $\lambda > 0$ 都成立. 进一步, p_i 的分子次数严格低于分母.

(P2): 格式(2.1)精确到 q 阶, 即

$$r(\lambda) = e^{-\lambda} + \mathcal{O}(\lambda^{q+1}), \quad \text{当 } \lambda \rightarrow 0.$$

另外, 对于每个 $0 \leq j \leq q$,

$$\sum_{i=1}^m c_i^j p_i(\lambda) - \frac{j!}{(-\lambda)^{j+1}} \left(e^{-\lambda} - \sum_{\ell=0}^j \frac{(-\lambda)^\ell}{\ell!} \right) = \mathcal{O}(\lambda^{q-j}), \quad \text{当 } \lambda \rightarrow 0.$$

(P3): 有理函数 $r(\lambda)$ 是强稳定的, 即 $|r(\infty)| < 1$.

例如, 可以参考专著 [27, p.131] 了解满足条件 (P1)-(P3) 的有理函数的构造. (P3) 对于确保并行迭代的收敛性至关重要, 特别是对于非光滑问题数据 (例如, $u^0 \in \mathcal{H}$). 当 $|r(\infty)| = 1$ (例如, Crank-Nicolson 和隐式 Runge-Kutta 的 Gauss 类型), 并行方法的收敛性取决于两个因素. 首先, \mathcal{A} 的特征值必须有上界, 这通常不是抛物方程的情况. 其次, 粗步长和细步长之间的比率必须足够大, 具体值由 \mathcal{A} 的特征值的上界决定. 不满足 (P3) 的格式可能会失去最优收敛率 [27, 第 8 章].

在实践中, 通常希望选择与 $r(\lambda)$ 有相同分母的 $p_i(\lambda)$, 即对于多项式 $a_i(\lambda)$ 和 $g(\lambda)$, 有

$$r(\lambda) = \frac{a_0(\lambda)}{g(\lambda)} \quad \text{和} \quad p_i(\lambda) = \frac{a_i(\lambda)}{g(\lambda)}, \quad \text{对于 } i = 1, 2, \dots, m.$$

然后格式(2.1)可以写成

$$g(\Delta t \mathcal{A}) u^{n+1} = a_0(\Delta t \mathcal{A}) u^n + \Delta t \sum_{i=1}^m a_i(\Delta t \mathcal{A}) f(t_n + c_i \Delta t), \quad 1 \leq n \leq N-1.$$

以下误差估计适用于格式 (2.1) ([27, 定理 7.2 和 8.1]).

引理 2.1. 假设条件 (P1)-(P3) 得到满足, $u(t)$ 是问题(1.1)的解, u^n 是格式(2.1)的解. 那么有

$$\|u^n - u(t_n)\| \leq c(\Delta t)^q \left(t_n^{-q} \|u^0\| + t_n \sum_{\ell=0}^{q-1} \sup_{s \leq t_n} \|\mathcal{A}^{q-\ell} f^{(\ell)}(s)\| + \int_0^{t_n} \|f^{(q)}(s)\| ds \right),$$

如果 $u_0 \in \mathcal{H}$, $f^{(\ell)} \in C([0, T]; \text{Dom}(\mathcal{A}^{q-\ell}))$ 对于 $0 \leq \ell \leq q-1$, 并且 $f^{(q)} \in L^1(0, T; \mathcal{H})$, 这里 $\text{Dom}(\mathcal{A})$ 代表 \mathcal{A} 的定义域.

注 2.1. 根据引理 2.1, 在条件 (P1)-(P3) 下, 如果 f 和 u_0 满足某些兼容条件, 格式(2.1)的解 u^n 会以 $\mathcal{O}((\Delta t)^q)$ 的阶数收敛到精确解 $u(t_n)$. 例如, 当 $\mathcal{A} = -\Delta$ 时, 配备零狄利克雷边界条件, 要求是 $(-\Delta)^\ell f^{(q-\ell)} = 0$ 在 $\partial\Omega$ 对于 $0 \leq \ell \leq q$. 为了规避这些严格条件, 假设格式(2.1)严格精确到 q 阶, 即

$$\sum_{i=1}^m c_i^j p_i(\lambda) - \frac{j!}{(-\lambda)^{j+1}} \left(r(\lambda) - \sum_{\ell=0}^j \frac{(-\lambda)^\ell}{\ell!} \right) = 0, \quad 0 \leq j \leq q-1.$$

如果截断误差对于所有 f 和 u_0 都为零, 使得解是 t 中至多 $p_0 - 1$ 阶的多项式, 则格式严格精确到 $p_0 < p$ 阶. 虽然具有给定 $m \in \mathbb{N}$ 的单步方法可能精确到 $2m$ 阶 (例如, *Gauss-Legendre* 方法, [9, 第 2.2 节]), 但它只能严格精确到最高 $m + 1$ 阶 [7, 引理 5].

注 2.2. 如果积分器表现出 L -稳定性, 即 $r(\infty) = 0$, 则可以改进引理 2.1 中的误差估计 ([27, 定理 7.2]).

2.2 并行算法

现在, 我们介绍单步积分器 (2.1) 的并行算法. 设 $\Delta T = J\Delta t$ 为粗步长, $N_c = T/\Delta T \in \mathbb{N}$, 并表示 $T_n = n\Delta T$. 数值传播器 \mathcal{G} 和 \mathcal{F} 分别对应粗和细时间网格. 通常, \mathcal{G} 是一种便宜的、低阶数值方法 (例如, 后向欧拉格式), 而 \mathcal{F} 由单步积分器 (2.1) 定义. 对于任何给定的初始数据 $v \in \mathcal{H}$ 和 $f \in C([0, T]; \mathcal{H})$, 粗和细网格传播器分别定义为

$$\mathcal{G}(T_n, \Delta T, v, f) = R(\Delta T \mathcal{A})v + \Delta T \sum_{i=1}^M P_i(\Delta T \mathcal{A})f(T_n + C_i \Delta T), \quad (2.2)$$

$$\mathcal{F}(t_n, \Delta t, v, f) = r(\Delta t \mathcal{A})v + \Delta t \sum_{i=1}^m p_i(\Delta t \mathcal{A})f(t_n + c_i \Delta t). \quad (2.3)$$

其中, M 和 m 分别表示粗网格传播器 \mathcal{G} 和细网格传播器 \mathcal{F} 的阶数, 以及 R 和 P 是粗网格传播器 \mathcal{G} 的有理函数, r 和 p 是细网格传播器 \mathcal{F} 的. 并行迭代在下面的算法中给出.

算法 1 并行算法.

1: **初始化:** 计算 $U_0^{n+1} = \mathcal{G}(T_n, \Delta T, U_0^n, f)$, 其中 $U_0^0 = u^0$, $n = 0, 1, \dots, N_c - 1$;

2: **for** $k = 0, 1, \dots, K$ **do**

3: 在每个子区间 $[T_n, T_{n+1}]$ 上, 依次计算 $j = 0, 1, 2, \dots, J - 1$

$$\tilde{U}_k^{n,j+1} = \mathcal{F}(T_n + j\Delta t, \Delta t, \tilde{U}_k^{n,j}, f),$$

初始值为 $\tilde{U}_k^{n,0} = U_k^n$, 并设 $\tilde{U}_k^{n+1} = \tilde{U}_k^{n,J}$.

4: 执行顺序校正, 即找到 U_{k+1}^{n+1} 通过

$$U_{k+1}^{n+1} = \mathcal{G}(T_n, \Delta T, U_{k+1}^n, f) + \tilde{U}_k^{n+1} - \mathcal{G}(T_n, \Delta T, U_k^n, f)$$

其中 $U_{k+1}^0 = u^0$, 对于 $n = 0, 1, \dots, N_c - 1$;

5: 检查停止准则.

并行算法的收敛性已经得到了广泛研究 (参见例如 [3, 6, 11, 14, 22, 28, 29, 32]). 最近, 针对线性演化方程, 杨、袁和周证明了使用后向欧拉格式作为 CP 时, 存在一个阈值 $J_* > 0$, 独立于 ΔT 、 Δt 和 \mathcal{A} , 即在条件 (P1)-(P3) 下, 如果 $J \geq J_*$, 那么以下界限成立, 其中 $\gamma \approx 0.3$,

$$\max_{1 \leq n \leq N_c} \|U_k^n - u^{nJ}\| \leq c\gamma^k. \quad (2.4)$$

这里, U_k^n 表示并行解, u^{nJ} 表示细时间步进解 ([32]). 数值上观察到估计 (2.4) 是精确的. 使用其他类型的 CP 时也会出现这个问题; 文献 [11] 报告了当 CP 和 FP 都是两阶、单对角隐式 Runge-Kutta 格式 (SDIRK-22) 时, 最优的网格独立收敛因子 $\gamma \approx 0.26$, 而当两者都是 SDIRK-33 时, $\gamma \approx 0.15$, 对于 $J \in \{4, 8, 16, 32, 64\}$. 这些观察自然地激发了以下问题: 是否存在可以显著提高收敛性的 CP, 同时在不同的 J 值上保持稳健性? 下面我们旨在学习新的 CP 来加速并行算法的收敛.

我们在实验中取 $\nu \in \{1, 0.02\}$. 对于 $\nu = 1$, 我们检验了 $J = 20, 100$ 和 $\Delta t = 1/2000$ (即 $\Delta T = 1/100$ 和 $1/20$) 的 LCPs. 结果显示, 使用 LCPs 的 Parareal 算法的收敛速率约为 0.04, 对于更具挑战性的情况 $\nu = 0.02$, 我们评估了 $J = 10, 100$ 和 $\Delta t = 1/10000$ (即 $\Delta T = 1/1000$ 和 $1/100$) 的 LCPs. 对于 $J = 10$, LCPs 的收敛速率约为 0.06, 而 BE 的收敛速率约为 0.21. 值得注意的是, 对于 $J = 100$ 在线性情况下, 比例 $\Delta T/\nu$ 保持不变, 因此它们的行为相似. 这一现象已经在 [13] 中对线性对流扩散案例进行了分析. 对于 $J = 100$, LCPs 和 BE 都表现出较慢的收敛.

第 3 章 总结

我们探索了在 Parareal 算法中学习粗网格传播器的潜力, 以克服收敛阶数的障碍. 这种方法基于对线性演化问题的严格误差估计, 确保了 LCPs 具有高度理想的特性, 例如稳定性和相容性. 我们为 CP 的学习开发了一个系统化的方法. 数值实验完全证实了理论预测, 包括轻微非线性问题. 最值得注意的是, 即使对于高度非线性问题, LCPs 仍然可以胜过经典的 BE 格式. 一个重要的问题是将这一思想扩展到更具挑战性的情况, 例如高度非线性问题、多尺度问题, 以及双曲问题 ([8, 20]). 此外, 为其他并行时间算法 (例如, 具有 FCF 平滑器的两级 MGRIT 算法 [11, 26]) 开发学习策略也很有前景.

参考文献

- [1] W. Agboh, O. Grainger, D. Ruprecht, and M. Dogar. Parareal with a learned coarse model for robotic manipulation. *Comput. Vis. Sci.*, 23(1-4):Paper No. 8, 10, 2020.
- [2] G. Ariel, S. J. Kim, and R. Tsai. Parareal multiscale methods for highly oscillatory dynamical systems. *SIAM J. Sci. Comput.*, 38(6):A3540–A3564, 2016.
- [3] G. Bal. On the convergence and the stability of the parareal algorithm to solve partial differential equations. In *Domain decomposition methods in science and engineering*, volume 40 of *Lect. Notes Comput. Sci. Eng.*, pages 425–432. Springer, Berlin, 2005.
- [4] G. Bal and Y. Maday. A “parareal” time discretization for non-linear PDE’s with application to the pricing of an American put. In *Recent developments in domain decomposition methods (Zürich, 2001)*, volume 23 of *Lect. Notes Comput. Sci. Eng.*, pages 189–202. Springer, Berlin, 2002.
- [5] Y. Bar-Sinai, S. Hoyer, J. Hickey, and M. P. Brenner. Learning data-driven discretizations for partial differential equations. *Proc. Natl. Acad. Sci. USA*, 116(31):15344–15349, 2019.
- [6] C.-E. Brehier and X. Wang. On parareal algorithms for semilinear parabolic stochastic PDEs. *SIAM J. Numer. Anal.*, 58(1):254–278, 2020.
- [7] P. Brenner, M. Crouzeix, and V. Thomée. Single-step methods for inhomogeneous linear differential equations in Banach space. *RAIRO Anal. Numér.*, 16(1):5–26, 1982.
- [8] H. De Sterck, R. D. Falgout, S. Friedhoff, O. A. Krzysik, and S. P. MacLachlan. Optimizing multigrid reduction-in-time and parareal coarse-grid operators for linear advection. *Numer. Linear Algebra Appl.*, 28(4):Paper No. e2367, 22, 2021.
- [9] B. L. Ehle. *On Padé approximations to the exponential function and A-stable methods for the numerical solution of initial value problems*. PhD thesis, University of Waterloo, Waterloo, Ontario, 1969.
- [10] S. Engblom. Parallel in time simulation of multiscale stochastic chemical kinetics. *Multiscale Model. Simul.*, 8(1):46–68, 2009.
- [11] S. Friedhoff and B. S. Southworth. On “optimal” h -independent convergence of parareal and multigrid-reduction-in-time using Runge-Kutta time integration. *Numer. Linear Algebra Appl.*, 28(3):Paper No. e2301, 30, 2021.
- [12] M. J. Gander. 50 years of time parallel time integration. In *Multiple shooting and time domain decomposition methods*, volume 9 of *Contrib. Math. Comput. Sci.*, pages 69–113. Springer, Cham, 2015.
- [13] M. J. Gander and T. Lunet. A Reynolds number dependent convergence estimate for the parareal algorithm. In *Domain decomposition methods in science and engineering XXV*, volume 138 of *Lect. Notes Comput. Sci. Eng.*, pages 277–284. Springer, Cham, 2020.
- [14] M. J. Gander and S. Vandewalle. Analysis of the parareal time-parallel time-integration method. *SIAM J. Sci. Comput.*, 29(2):556–578, 2007.
- [15] D. Greenfeld, M. Galun, R. Basri, I. Yavneh, and R. Kimmel. Learning to optimize multigrid PDE solvers. In *Proceedings of the International Conference on Machine Learning*, pages 2415–2423, 2019.

- [16] Y. Guo, F. Dietrich, T. Bertalan, D. T. Doncevic, M. Dahmen, I. G. Kevrekidis, and Q. Li. Personalized algorithm generation: a case study in learning ODE integrators. *SIAM J. Sci. Comput.*, 44(4):A1911–A1933, 2022.
- [17] R. Huang, R. Li, and Y. Xi. Learning optimal multigrid smoothers via neural networks. *SIAM J. Sci. Comput.*, 45(3):S199–S225, 2023.
- [18] A. Q. Ibrahim, S. Götschel, and D. Ruprecht. Parareal with a physics-informed neural network as coarse propagator. In J. Cano, M. D. Dikaiakos, G. A. Papadopoulos, M. Pericás, and R. Sakellariou, editors, *Euro-Par 2023: Parallel Processing*, pages 649–663, 2023.
- [19] B. Jin, Q. Lin, and Z. Zhou. Learning coarse propagators in parareal algorithm. *arXiv preprint arXiv:2311.15320*, 2023.
- [20] G. Li and J. Hu. Wavelet-based edge multiscale parareal algorithm for parabolic equations with heterogeneous coefficients and rough initial data. *J. Comput. Phys.*, 444:10572, 18, 2021.
- [21] J.-L. Lions, Y. Maday, and G. Turinici. Résolution d’EDP par un schéma en temps “pararéel”. *C. R. Acad. Sci. Paris Sér. I Math.*, 332(7):661–668, 2001.
- [22] T. P. Mathew, M. Sarkis, and C. E. Schaerer. Analysis of block parareal preconditioners for parabolic optimal control problems. *SIAM J. Sci. Comput.*, 32(3):1180–1200, 2010.
- [23] S. Mishra. A machine learning framework for data driven acceleration of computations of differential equations. *Math. Eng.*, 1(1):118–146, 2019.
- [24] H. Nguyen and R. Tsai. Numerical wave propagation aided by deep learning. *J. Comput. Phys.*, 475:Paper No. 111828, 26, 2023.
- [25] G. Pagès, O. Pironneau, and G. Sall. The parareal algorithm for American options. *C. R. Math. Acad. Sci. Paris*, 354(11):1132–1138, 2016.
- [26] B. S. Southworth. Necessary conditions and tight two-level convergence bounds for parareal and multigrid reduction in time. *SIAM J. Matrix Anal. Appl.*, 40(2):564–608, 2019.
- [27] V. Thomée. *Galerkin finite element methods for parabolic problems*. Springer-Verlag, Berlin, second edition, 2006.
- [28] S.-L. Wu. Convergence analysis of some second-order parareal algorithms. *IMA J. Numer. Anal.*, 35(3):1315–1341, 2015.
- [29] S.-L. Wu and T. Zhou. Convergence analysis for three parareal solvers. *SIAM J. Sci. Comput.*, 37(2):A970–A992, 2015.
- [30] Q. Xu, J. S. Hesthaven, and F. Chen. A parareal method for time-fractional differential equations. *J. Comput. Phys.*, 293:173–183, 2015.
- [31] G. R. Yalla and B. Engquist. Parallel in time algorithms for multiscale dynamical systems using interpolation and neural networks. In *HPC ’18: Proceedings of the High Performance Computing Symposium*, pages 9, 1–12, 2018.
- [32] J. Yang, Z. Yuan, and Z. Zhou. Robust convergence of parareal algorithms with arbitrarily high-order fine propagators. *CSIAM Trans. Appl. Math.*, 4(3):566–591, 2023.

致 谢

在此，我要感谢我的导师和同事对我的支持和帮助。

附录

A 显式公式

在图 1 中

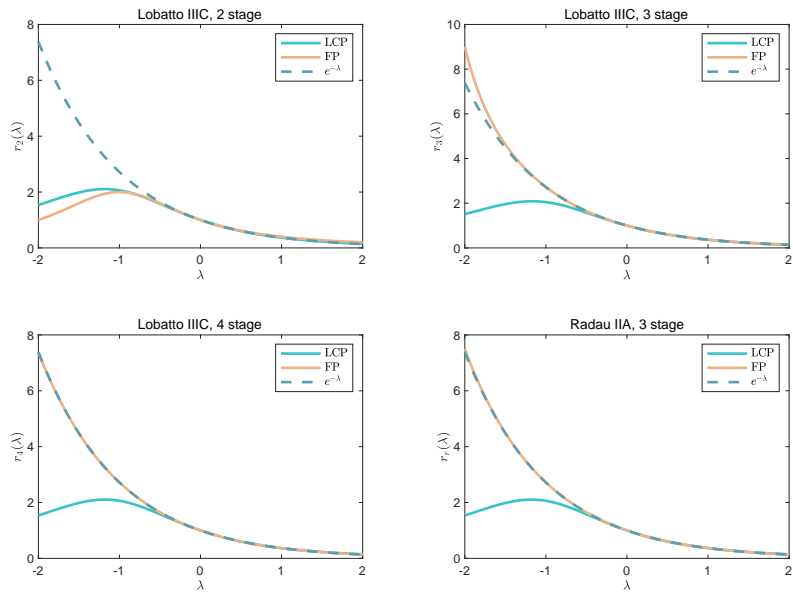


图 1: 四种 FPs 的 LCPs 的稳定函数.