微分方程初值问题的并行数值方法

郭嘉

摘要

本文首先分析了微分方程初值问题并行算法设计的困难性，然后总结了部分相关的并行算法，并针对他们的适用情形作出了讨论。

1 初值问题并行方法的本质困难性

微分方程的初值问题在物理、化学、计算生物学以及工程和金融领域都有十分重要的作用。本文主要考虑以下常微分方程的初值问题：

$$y'(x) = f(x, y(x)), x \in [x_0, x_T] \tag{1}$$

$$y(x_0) = y_0 \tag{2}$$

以上常微分方程在很多情况下没有解析解，只能通过数值方法进行求解，然而不论是常用的单步法还是多步法，其计算过程都是串行的。这可以通过发展方程的物理意义进行解释：给定某物理量的初始状态以及该状态的变化与状态和时间之间的关系，我们通常只能顺着时间的发展观察其变化。

为了说明初值问题数值方法并行化的困难性，

我们以最简单的显式欧拉法为例：

$$y_{n+1} = y_n + hf(x_n, y_n) \tag{3}$$

容易看出，为了求得 y_{n+1} 我们必须先求出 y_n 的值。单步法走完前一步再走后一步的串行性是很直观的，事实上，对于常用的四阶龙格-库塔法来说，即使是每步之内都是串行的：

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{4}$$

其中

$$k_1 = f(x_n, y_n) \tag{5}$$

$$k_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1) \tag{6}$$

$$k_3 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2) \tag{7}$$

$$k_4 = f(x_n + h, y_n + hk_3) \tag{8}$$

从上面容易看出为了求出 k_{i+1} 我们必须先得到 k_i，人们自然想到了并行龙格-库塔算法的设计，但是并行的代价在于方法的稳定性和截断误差阶数会受到一定限制。在本文的第二部分我们会总结一些并行龙格-库塔算法的工作。

我们下面将目光转向线性多步法，多步法的预估-校正 (P-C) 步明显是串行的，我们以最简单的 P-C 法为例：

$$y^p_{n+1} = y^c_n + \frac{h}{2}(3f(x_n, y^c_n) - f(x_n, y^c_{n-1})) \tag{9}$$

$$y^c_{n+1} = y^c_n + \frac{h}{2}(3f(x_{n+1}, y^p_{n+1}) - f(x_n, y^c_n)) \tag{10}$$

可以看出为了求出 y^c_{n+1} 我们必须得到 y^p_{n+1}, 为了求出 y^p_{n+1} 我们必须得到 y^c_n，这两者间一环紧扣一环，这乍一看似乎很难设计合适的并行算法。本文的第二部分将会总结一些相关的工作。

我们在偏微分方程中的典型初值问题为热传导方程的初值问题，这里考虑一维形式, 且 u 满足齐次边值条件 (下面不列出)：

$$u_t = u_{xx}, 0 < x < 1, t > 0 \tag{11}$$

$$u(x, 0) = \varphi(x) \tag{12}$$

与常微分方程初值问题不同，这里我们不仅在时间上可以考虑并行，同时可以考虑如何在空间上进行并行，本文的第二部分会总结相关的方法。

2　微分方程初值问题并行化算法概述

2.1　粗预估-时间并行算法

早期，Nievergelt[Nie64] 为了克服(3)中为了计算某个时间点的变化率必须知道在该点处的函数值的困难，提出先通过简单的串行数值方法（如显式欧拉法）估计出在每个节点 x_i 处的函数值 y_{i0}，然后在每个 y_{i0} 附近取 M_i 个点 y_{iM_i}，通过并行计算出这些点处的导数值对原来的预估值进行修正从而得到最终的估计值 $\tilde{y}_i$。

该算法是 Nievergelt 在 1964 年提出的，他当时就洞见到了数值计算问题并行算法研究的重要性，他指出电子元件约束了计算机能达到的最高算力，未来许多数值计算方法都会对并行性给予更大的关注。

2.2　迭代并行算法

Bellen 和 Zennaro [BZ89] 试图将单步法迭代式(13)在时间维度上进行拼接成(14)(其中 $\mathbf{y} = (y_0, y_1, ..., y_N)$)：

$$y_{n+1} = \varphi(y_n) \tag{13}$$

$$\mathbf{y} = \phi(\mathbf{y}) \tag{14}$$

这样就把一个串行的单步法转换成了可以并行求解的非线性方程组问题，利用牛顿法或史蒂芬森法进行求解都能够进行并行计算。

事实上，迭代算法可以不用拘泥于如(13)的单步方法，Saha 等人 [SST96] 考虑以下的近似:

$$y_n = y_0 + h\sum_{m=0}^{n-1} f(\frac{1}{2}(y_{m+1}+y_m)) \tag{15}$$

不难看出，通过(15)的格式我们也能够构造出(14)的迭代表达。

我们知道，利用牛顿法进行迭代需要初值十分接近目标解，否则可能导致发散。我们只需要利用简单串行算法进行初步的估计作为初始值即可。牛顿法所涉及的矩阵求逆、雅可比矩阵计算都是并行计算机可以极大提速的地方。且牛顿法在目标解附近是 2 阶收敛的，因此我们能够期望(14)的求解在并行计算机下是很快的。

2.3 龙格-库塔法的并行

我们在本文的第一部分看到，常用的四阶龙格-库塔格式是不能并行的。容易看到，只有当龙格-库塔法的系数矩阵 $A \in R^{s\times s}$ 能够被分成 $p\times p$ 块的分块矩阵时，我们才能考虑进行 p 步串行，且在每步内并行的方法:

$$
A = \begin{pmatrix}
D_1 & 0 & 0 & \cdots & 0 \\
A_{21} & D_2 & 0 & \cdots & 0 \\
A_{31} & A_{32} & D_3 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & 0 \\
A_{p1} & A_{p2} & \cdots & \cdots & A_{pp}
\end{pmatrix} \tag{16}
$$

对于显式的龙格-库塔算法我们要求 D_i 是零矩阵，对于隐式的龙格库塔算法，我们要求 D_i 是对角矩阵。[JN95] 和 [VDHS90] 给出了一些显式的方法，但同时也指出该方法的收敛阶不超过 p 阶，并且当其能够达到 p 阶收敛时，其绝对稳定域也很小。[IN90] 提出了一种隐式方法，系数矩阵为:

$$
A = \begin{pmatrix}
1/2 & 0 & 0 & 0 \\
0 & 2/3 & 0 & 0 \\
-5/2 & 5/2 & 1/2 & 0 \\
-5/3 & 4/3 & 0 & 2/3
\end{pmatrix} \tag{17}
$$

从上面矩阵可以看出其可以在 2 个处理器上进行并行计算。而且其收敛阶数为 4, 并且是 L-稳定的。

2.4 预估-校正法的并行

回顾第一部分中给出的(9)(10)，我们知道预估-校正法的设计上本质是串行的，[ML67][Mir71] 将(9)替

换成了下式，容易看出，这实际上给出了一种预估-校正法的二处理器并行版本：

$$y^p_{n+1} = y^c_{n-1} + 2hf(x_n, y^p_n) \tag{18}$$

2.5 区域分解法

为了使用并行算法解决(11)和(12)的热传导方程初值问题，[Gan96] 提出了一种与 Schwarz 区域分解法类似的算法，该算法在空间上取得了并行：

$$\partial_t u^n_1 = \partial_{xx} u^n_1, 0 < x < \beta, t > 0 \tag{19}$$

$$u^n_1(\beta, t) = u^{n-1}_2(\beta, t) \tag{20}$$

$$\partial_t u^n_2 = \partial_{xx} u^n_2, \alpha < x < 1, t > 0 \tag{21}$$

$$u^n_2(\alpha, t) = u^{n-1}_1(\alpha, t) \tag{22}$$

其中 $\alpha < \beta$。

3 讨论

在第二部分提出的算法中，我们可以将其分为两大类，2.1-2.4 节提出的算法在时间维度上取得并行，2.5 节提出的算法在空间上取得并行。在第一大类

的算法中又可以分为两个大类，2.1，2.2 节的并行算法属于大规模的并行，其能支持的并行处理器数至少为 $O(n)$，2.3，2.4 节的并行算法属于小规模的并行，其能支持的并行处理器数为 $O(1)$。如果处理器数量较多，但个别处理器的处理性能不佳时，可以考虑采用大规模并行，反之采用小规模并行。

由于有时需要通过迭代求解方程，隐式的数值方法在串行机器上会比较耗时。但出于稳定性和收敛阶方面的考虑，我们也会使用。但如果允许大规模的并行，隐式的方法如隐式龙格-库塔法将会有很大的优势。

References

[BZ89] Alfredo Bellen and Marino Zennaro. Parallel algorithms for initial-value problems for difference and differential equations. *Journal of Computational and applied mathematics*, 25(3):341–350, 1989.

[Gan96] Martin Jakob Gander. Overlapping schwarz for linear and nonlinear parabolic problems. In *9th International Conference on Domain Decomposition Methods*, pages 97–104, 1996.

[IN90] Arieh Iserles and SP Nørsett. On the theory of parallel runge—kutta methods. *IMA Journal of numerical Analysis*, 10(4):463–488, 1990.

[JN95] Kenneth R Jackson and Syvert Paul Nørsett. The potential for parallelism in runge–kutta methods. part 1: Rk formulas in standard form. *SIAM journal on numerical analysis*, 32(1):49–82, 1995.

[Mir71] WL Miranker. A survey of parallelism in numerical analysis. *Siam Review*, 13(4):524–547, 1971.

[ML67] Willard L Miranker and Werner Liniger. Parallel methods for the numerical integration of ordinary differential equations. *Mathematics of Computation*, 21(99):303–320, 1967.

[Nie64] Jürg Nievergelt. Parallel methods for integrating ordinary differential equations. *Communications of the ACM*, 7(12):731–733, 1964.

[SST96] Prasenjit Saha, Joachim Stadel, and Scott Tremaine. A parallel integration method for solar system dynamics. *arXiv preprint astro-ph/9605016*, 1996.

[VDHS90] Peter J Van Der Houwen and Benjamin P Sommeijer. Parallel iteration of high-order runge-kutta methods with stepsize control. *Journal of Computational and Applied Mathematics*, 29(1):111–127, 1990.