arXiv:2512.13845v1 [cs.CE] 15 Dec 2025

步长变化导致的协同仿真误差

Lars T. Kyllingstad*

摘要

在连续时间协同仿真中，如果两个仿真单元通过某个变量 q相连，且两个仿真单元都具有一个表示 q时间积分的内部状态，那么由于外推误差，这些状态通常会出现偏差。通常，减小宏观时间步长可以降低此类外推误差。本文将表明，在某些情况下，步长变化会导致此类偏差增大，即使步长变得更小也是如此。

1 引言

协同仿真是一种用于仿真耦合动力系统的方法，其做法是将所关注的系统划分为多个子系统，各子系统彼此独立地进行仿真，仅在离散的通信点交换信息。在通信点之间，每个仿真单元 负责推进自身内部状态的时间演化，例如通过对一组常微分方程进行积分。

子系统的这种解耦使协同仿真特别适用于多物理场仿真、分布式计算，以及在工业环境中保护各个子系统的敏感信息 [1]。然而，这也会带来误差。由于仿真单元只能在通信点采样其输入的真实值，因此必须对两次通信之间的时间间隔内的输入值进行近似。采用显式协同仿真方法时，仿真单元必须根据过去的输入值进行外推；采用隐式方法时，则可以在相邻通信点之间对输入变量进行插值。这类近似本质上并不精确，可能导致通信点处出现不连续性。因此，协同仿真除了子系统行为数值计算中已有的误差外，还会引入耦合误差 。

只要仿真单元收敛且其行为连续，耦合误差就会按某个幂次随通信点 [2–5]，之间的时间间隔缩放，这个间隔称为宏观时间步长。[1] 这意味着，可以通过缩短步长来减少误差，但代价是计算时间增加。变步长协同仿真算法便利用这一点，在仿真过程中动态调整步长，以在将误差控制在一定范围内的同时最大限度地提升性能 [5–11]。

*SINTEF Ocean，特隆赫姆，挪威。*lars.kyllingstad@sintef.no*。ORCID 0000-0002-4334-4490。[1]使用前缀“宏”是为了将通信点之间的时间步长与仿真单元内部使用的（微观）时间步长区分开来。本文完全不会提及微观时间步长，因此通常会省略该前缀。

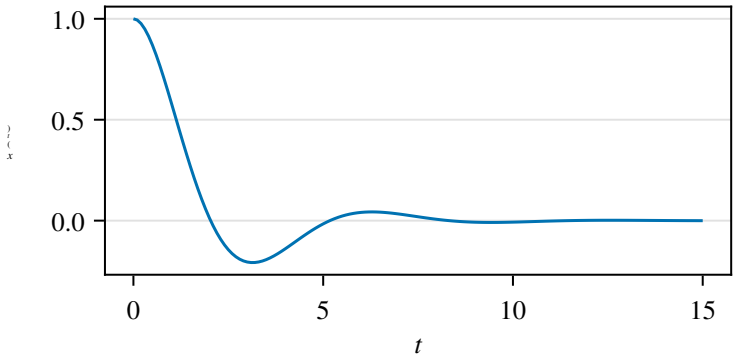

图1：阻尼谐振子的解，其中 $m = c = k = 1$ 和初始条件 $x(0) = 1$ 以及 $\dot{x}(0) = 0$。

然而，正如本文将要阐明的，在某些情况下，仅仅改变步长这一操作就可能引入误差，即使步长变得更短也是如此。具体而言，我们将说明，当一个耦合变量在连接两侧都进行积分，导致一侧的积分基于近似值时，由此产生的耦合误差与连续步长之间的差值 成正比。

有人可能会认为，在一次仿真中包含同一状态的两个实例是多余的，应当避免。为了提高数值精度，最好只在一个仿真单元中维护积分状态，并将其作为输出公开，以便其他仿真单元访问其“真实”值。然而，在协同仿真中，使仿真单元尽可能自包含、避免对其他仿真单元产生结构依赖，从而确保模块化、可复用性和可扩展性，通常是完全合理的。为其他仿真单元提供最小化接口正是其中 [11]的一部分。

所述情况通常发生在仿真单元之间交换一个流量变量——例如速度、角速度、电流或体积流量——并对其进行积分以生成位移——分别为位置、角度、电荷或排出体积时。本文将在第2节和第4节介绍两个此类示例，并在第3节中对这一现象进行一般性推导。最后，我们将在第5节讨论如何减轻所述误差，并以一些观察结论结束本文。

2 示例1：阻尼谐振子

作为第一个示例，我们考虑阻尼谐振子的情况：

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = 0. \tag{1}$$

为清晰起见，我们采用无量纲单位，并令质量、阻尼系数和弹簧常数均为 1：$m = c = k = 1$。图1展示了初始条件 $x(0) = 1$ 和 $\dot{x}(0) = 0$下的解。

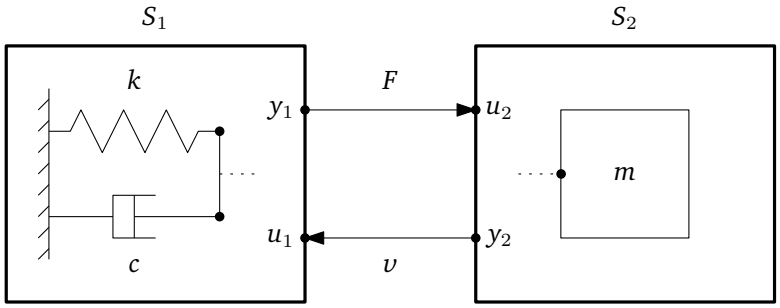

图 2：阻尼谐振子的协同仿真设置。

为进行协同仿真，我们将其视为一个质量–弹簧–阻尼器系统，并按照图2所示将其拆分为两个仿真单元。仿真单元 S_1 代表弹簧和阻尼器。它有一个输入 u_1，即弹簧和阻尼器被拉伸的速度；一个输出 y_1，即它们抵抗拉伸所施加的力；以及一个内部状态 x_1，即它们被拉伸的距离。因此， S_1 的状态方程为

$$\dot{x}_1(t) = \tilde{u}_1(t). \tag{2}$$

波浪号表示外推，这是必要的，因为每个仿真单元只能在离散通信点处知道其输入变量的真实值$t[0]$， $t[1]$， ……， $t[n]$。本文始终假设采用最简单的外推方案，即保持值不变（即零阶保持）：

$$\tilde{u}_1(t) = u_1[n], \qquad t[n] < t \le t[n+1]. \tag{3}$$

尖括号表示离散时间，也就是$\cdot[n] \equiv \cdot(t[n])$。因此， S_1 的输出方程写为

$$y_1[n] = -kx_1[n] - c\tilde{u}_1[n]. \tag{4}$$

仿真单元 S_2 表示质量。它还有一个输入 u_2，即作用于质量的力；一个输出 y_2，即其速度；以及两个内部状态x_2 和 v_2，分别表示其位移和速度。因此， S_2 的状态空间方程为：

$$\dot{x}_2(t) = v_2(t), \tag{5}$$

$$\dot{v}_2(t) = \frac{1}{m}\tilde{u}_2(t), \tag{6}$$

$$y_2[n] = v_2[n]. \tag{7}$$

在通信点处实施的耦合约束为：

$$u_1[n] = y_2[n] \qquad \text{(velocity)}, \tag{8}$$

$$u_2[n] = y_1[n] \qquad \text{(force)}. \tag{9}$$

鉴于我们采用零阶保持，两个仿真

参数	符号	值
初始步长	$\Delta t[0]$	0.1
比例增益	k_P	0.2
积分增益	k_I	0.1
最小步长	$\Delta t_{\min}$	10^{-5}
最大步长	$\Delta t_{\max}$	0.1
最大相对步长缩减	$\theta_{\min}$	0.2
最大相对步长增加	$\theta_{\max}$	1.2
绝对容差	δ	10^{-6}
相对容差	σ	10^{-6}

表1：变步长算法和误差估计器的参数。

单元的状态方程很容易通过解析方法求解，因此得到：

$$x_1[n+1] = x_1[n] + u_1[n]\Delta t[n], \tag{10}$$

$$x_2[n+1] = x_2[n] + v_2[n]\Delta t[n] + \frac{1}{2m}u_2[n]\Delta t[n]^2, \tag{11}$$

$$v_2[n+1] = v_2[n] + \frac{1}{m}u_2[n]\Delta t[n], \tag{12}$$

其中， $\Delta t[i] \equiv t[i+1] - t[i]$为宏观时间步长。

随后，我们使用两种不同的协同仿真算法来仿真该系统：一种是固定步长算法，对所有 i均采用 $\Delta t[i] = \Delta t_{\text{fixed}} = 0.1$；另一种是变步长算法。对于后一种情况，我们使用 Kyllingstad、Sadjina 和 Skjong [11] 所述的 PI 控制算法，并采用基于基于能量守恒的协同仿真（ECCO）方法的误差估计 [8]。为确保完整性和可复现性，表1列出了 PI控制器和误差估计器所用的参数；不过，算法或误差估计器的具体选择在这里几乎无关紧要，因此我们不再深入讨论细节。

由于我们有子系统在通信点之间行为的解析表达式（10–12），因此无需在仿真单元内进行数值积分；接下来观察到的任何误差都必然完全是协同仿真的产物。我们也规避了积分与宏步长之间不恰当的相互作用可能引发的任何问题（例如，参见 Skjong 和 Pedersen [12]）。

2.1 谜题

图3展示了两次仿真的结果。首先可以看到，整体结果（左侧图）彼此之间以及与图1所示的精确解都相当相似。不过，放大查看尾部（右侧图）后，我们发现位移状态 x_1 和 x_2 之间存在偏差，这种偏差在变步长情形下尤为明显。

仿真单元之间存在偏差本身并不令人意外。位移是通过对速度进行积分得到的；在 S_1中，来自 S_2 的速度信号经过采样后，使用零阶保持器进行重构，但重构并不准确。令人意外的是， x_1 与 x_2 之间的差异显著

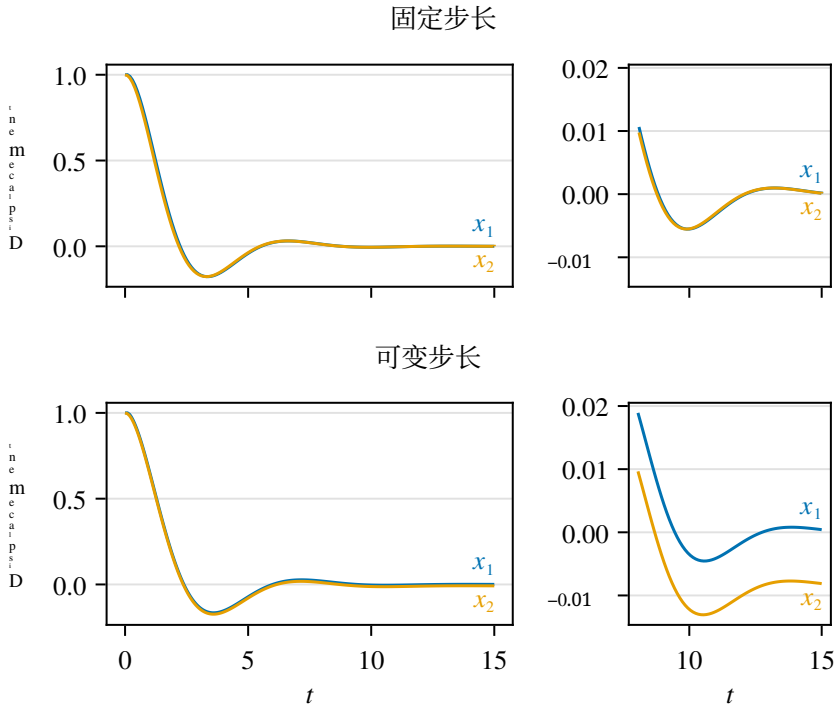

图3：谐振子的协同仿真解。上方的图展示了采用固定步长获得的解；下方的图对应变步长算法。最左侧的图展示了整个仿真过程；最右侧的图则放大显示了尾部（$9 \le t \le 15$）。

更大，尽管我们已将控制器配置为使用与固定步长算法相同或更小的宏步长（$\Delta t_{\max} = \Delta t_{\text{fixed}}$）。

为了弄清这里发生了什么，我们更仔细地考察 S_1 和 S_2中的位移状态之间的偏差，

$$\Delta x[n] \equiv x_1[n] - x_2[n], \tag{13}$$

同时比较两次仿真的步长，结果如图4所示。在固定步长情况下，我们看到偏差在仿真的第一阶段初始时急剧增加，并且变化很大。这是意料之中的，因为就速度信息而言， S_1 滞后于 S_2 ，而速度在这一阶段也变化很大。然而，随着系统趋于静止，偏差逐渐消失。

在变步长情形中，我们从相同的初始步长开始，在最初几个时间步内，两条曲线几乎始终保持一致。在某一点，步长算法将步长减小为较小的 Δt，此时偏差不再急剧增加，而是进入振荡幅度小得多的阶段，随后也趋于稳定。然而，最终稳定在一个更大的

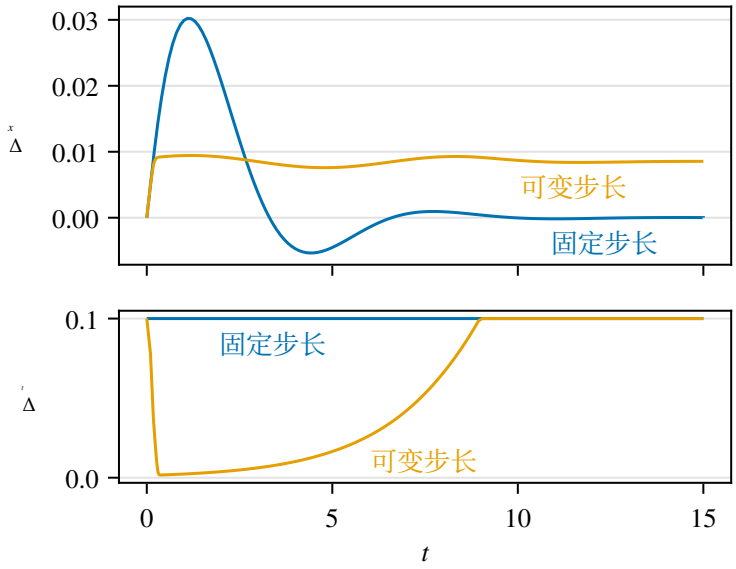

图4：上方面板显示了在 S_1 和 S_2、 $\Delta x = x_1 - x_2$中，固定步长和可变步长情形下位移状态之间的偏差。下方面板显示了宏观时间步长。仿真以初始速度 $\dot{x}_1 = \dot{x}_2 = 0$运行。

值上，而不是固定步长情形中的值。早期阶段的较大偏差似乎因步长的突然减小而被“冻结”了。

显然，我们观察到了一种似乎与步长相关的数值误差，但其表现方式并不寻常：它似乎并非与步长大小相关，也不会随着步长减小而减小，而是与步长变化相关。不过，它似乎还严重依赖于系统状态，如图5所示。在这里，我们使用初始速度 $\dot{x}_1[0] = \dot{x}_2[0] = 1$重新运行了仿真，现在可以看到两者的表现发生了互换；在这种情况下，可变步长算法表现更好。

2.2 解释

请注意，子模拟器内部状态的方程 10–12 都采用递推关系的形式。将这些方程反复应用，直到 $n = 0$，可得

$$x_1[n] = x_1[0] + \sum_{i=0}^{n-1} u_1[i]\Delta t[i], \tag{14}$$

$$x_2[n] = x_2[0] + \sum_{i=0}^{n-1} \left(v_2[i]\Delta t[i] + \frac{1}{2m} u_2[i]\Delta t[i]^2 \right), \tag{15}$$

$$v_2[n] = v_2[0] + \frac{1}{m} \sum_{i=0}^{n-1} u_2[i]\Delta t[i]. \tag{16}$$

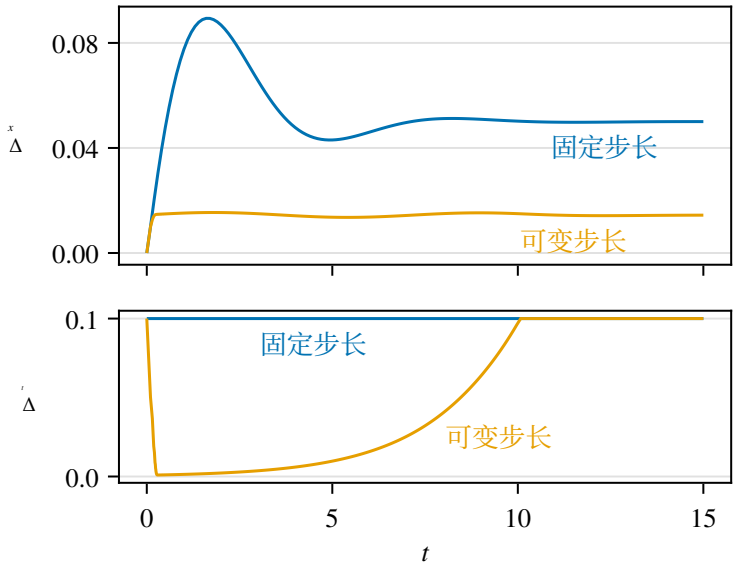

图5：上方面板显示了 S_1 中位移状态在 S_2、 $\Delta x = x_1 - x_2$下固定步长和可变步长情形之间的偏差。下方面板显示宏步长。与图4不同，仿真采用了初始速度 $\dot{x}_1 = \dot{x}_2 = 1$。

如果我们假设两个位移状态均初始化为相同的值（$x_1[0] = x_2[0]$），并利用输出方程 (7) 和约束方程 (8)，二者共同意味着

$$u_1[i] = y_2[i] = v_2[i], \tag{17}$$

然后，我们得到状态偏差（13）的以下紧凑表达式：

$$\Delta x[n] = -\frac{1}{2m}\sum_{i=0}^{n-1} u_2[i]\Delta t[i]^2. \tag{18}$$

这是我们在引言中讨论过的那类外推误差的一个例子。通常的处理方法有：

- 传递耦合变量的高阶时间导数。对于这个特定系统，如果将 $\dot{u}_1[n] = \dot{y}_2[n]$加入耦合约束， $\Delta t[i]^2$ 阶项也会完全抵消。
- 如果无法做到这一点，则在 S_1 内进行高阶输入外推，以获得近似值 $\dot{y}_2$、 $\ddot{y}_2$等。
- 减小步长，因为显然，对于所有 $i \in \{0, 1, \ldots\ldots, n-1\}$， $\Delta x[n] \to 0$ 随 $\Delta t[i] \to 0$呈二次变化。

最后一种情况最简单，但正如我们所见，事情显然并非如此简单。

首先考虑 $\Delta t[i] = \Delta t_{\text{fixed}} = \text{const}$ 的固定步长情形。此时，我们可以写成

$$\Delta x[n] = -\frac{1}{2m}\Delta t_{\text{fixed}} \sum_{i=0}^{n-1} u_2[i]\Delta t_{\text{fixed}} = -\frac{1}{2}(v_2[n] - v_2[0])\Delta t_{\text{fixed}}, \tag{19}$$

其中，我们在上一步中使用了表达式 (16) 来表示 $v_2[n]$。这是一个耗散系统，最终会静止下来，因此当 $n \to \infty$ 时，$v_2[n] \to 0$。因此，最终我们会得到一个恒定误差：

$$\Delta x[\infty] = \frac{1}{2}v_2[0]\Delta t_{\text{fixed}}. \tag{20}$$

这正是我们在图4和图5的上方图表中看到的。

接下来，我们转向变步长情形，并暂且假设只在通信点 K 处改变一次步长：

$$\Delta t[i] = \begin{cases} \Delta t_1, & i < K \\ \Delta t_2, & i \geq K \end{cases} \tag{21}$$

对于 $n \geq K$，位移差异现在变为

$$\Delta x[n] = -\frac{1}{2m}\Delta t_1 \sum_{i=0}^{K-1} u_2[i]\Delta t_1 - \frac{1}{2m}\Delta t_2 \sum_{i=K}^{n-1} u_2[i]\Delta t_2 \tag{22}$$

$$= -\frac{1}{2}(v_2[K] - v_2[0])\Delta t_1 - \frac{1}{2}(v_2[n] - v_2[K])\Delta t_2 \tag{23}$$

$$= \frac{1}{2}v_2[0]\Delta t_1 + \frac{1}{2}v_2[K](\Delta t_2 - \Delta t_1) - \frac{1}{2}v_2[n]\Delta t_2. \tag{24}$$

最后一个表达式与定步长情形下的式 (19) 类似，但现在多了一项由步长变化引起的贡献，该项与两个步长之差成正比。随着仿真进行，系统最终静止下来，我们得到最终偏差：

$$\Delta x[\infty] = \frac{1}{2}v_2[0]\Delta t_1 + \frac{1}{2}v_2[K](\Delta t_2 - \Delta t_1). \tag{25}$$

如果只考虑仿真开始时步长的突然变化，并忽略此后步长的缓慢增加，我们也可以在结果中看到这一点。图4中，步长在 $t[K] \approx 0。2$ 处下降了一个数量级，因此

$$\Delta t_2 - \Delta t_1 \approx -\Delta t_1 = -0.1. \tag{26}$$

在同一点，速度（未绘出）为 $v_2[K] \approx -0。2$。因此，在这种情况下，我们得到 $\Delta x[\infty] \approx 0。1$，这与图中所示相当接近。

对于另一个初始条件 $v_2[0] = 1$，这种单步长变化简化方法的效果不太理想，这可能是因为在早期宏观时间步期间，v_2 的值较大。下一节中，我们将推广到任意数量的步长变化。

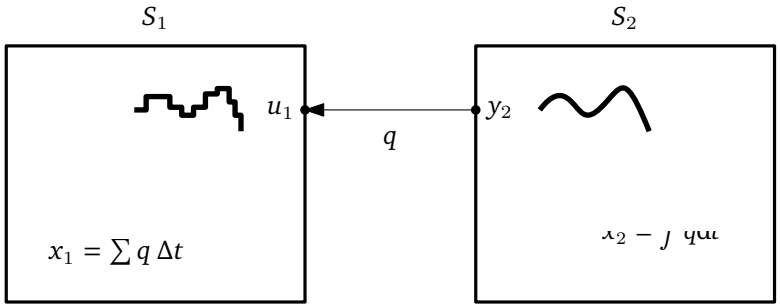

图 6：两个系统通过流量相连，流量在两侧均会累积。

3 一般情形

现在考虑一般情形：有两个未指定的系统 S_1 和 S_2，其中 S_1 有一个输入变量 u_1，该变量连接到 S_2的输出变量 y_2。（请注意，虽然我们的第一个示例基于闭环连接，但这与此处讨论的问题无关。）我们用 q 表示通过此连接传输的流量，因此在同步点 n，

$$u_1[n] = y_2[n] = q[n] = q(t[n]). \tag{27}$$

两个子系统中都有状态，分别表示 $q(t)$ 的积分，记作 x_1 和 x_2。该结构如图 6 所示。

在 S_2 中， $q(t)$ 完全已知，状态空间方程为

$$\dot{x}_2(t) = q(t), \tag{28}$$
$$y_2[n] = q[n]. \tag{29}$$

在 S_1 中， q 由输入变量 u_1 表示，相应的状态方程为

$$\dot{x}_1(t) = \tilde{u}_1(t) = u_1[n] = y_2[n] = q[n], \qquad t[n] < t \le t[n+1], \tag{30}$$

这里我们再次使用零阶保持法对输入值进行了外推。在通信点 n处， x_1 的值为

$$\begin{aligned} x_1[n] &= x_1[n-1] + \int_{t[n-1]}^{t[n]} \mathrm{d}t\ \tilde{u}_1(t) \\ &= x_1[n-1] + q[n-1]\Delta t[n-1] \\ &= x_1[0] + \sum_{i=0}^{n-1} q[i]\Delta t[i]. \end{aligned} \tag{31}$$

最后一行是通过反复应用第二行中的递推关系得出的。

现在，我们将推导一个结构类似的表达式，用于 x_2。我们假设函数 $q(t)$ 在 $t[n]$ 处实解析，且其泰勒级数收敛至 $q(t)$

对所有 $t \in [t[n], t[n+1]]$ 以及所有通信点 n 均成立。换句话说，我们可以写成

$$q(t) = \sum_{k=0}^{\infty} \frac{1}{k!} q^{(k)}[n-1](t - t[n])^k, \qquad t[n] \le t \le t[n+1], \tag{32}$$

其中， $q^{(k)}[i]$ 表示 $q(t)$ 在 $t = t[i]$处的 k^{th} 导数。因此可得

$$\begin{aligned} x_2[n] &= x_2[n-1] + \int_{t[n-1]}^{t[n]} \mathrm{d}t\ q(t) \\ &= x_2[n-1] + \sum_{k=0}^{\infty} \frac{1}{k!} q^{(k)}[n-1] \int_{t[n-1]}^{t[n]} \mathrm{d}t\ (t - t[n-1])^k \\ &= x_2[n-1] + \sum_{k=0}^{\infty} \frac{1}{(k+1)!} q^{(k)}[n-1] \Delta t[n-1]^{k+1} \\ &= x_2[0] + \sum_{i=0}^{n-1} \sum_{k=0}^{\infty} \frac{1}{(k+1)!} q^{(k)}[i] \Delta t[i]^{k+1}. \end{aligned} \tag{33}$$

现在可以将 x_1 与 x_2 （在通信点处）之间的偏差表示为

$$\begin{aligned} \Delta x[n] &\equiv x_1[n] - x_2[n] \\ &= x_1[0] - x_2[0] + \sum_{i=0}^{n-1} \left(q[i] \Delta t[i] - \sum_{k=0}^{\infty} \frac{1}{(k+1)!} q^{(k)}[i] \Delta t[i]^{k+1} \right) \\ &= \Delta x[0] - \sum_{i=0}^{n-1} \sum_{k=1}^{\infty} \frac{1}{(k+1)!} q^{(k)}[i] \Delta t[i]^{k+1}. \end{aligned} \tag{34}$$

为了更清楚地看出与步长变化的联系，我们再作两个假设：首先，状态 x_1 和 x_2 具有相同的初始值，因此 $\Delta x[0] = 0$；其次，所有步长都足够小，使得 $t[n]$充分位于 $q(t)$ 在 $t[n-1]$ 处的泰勒级数的收敛半径内，因此可以忽略 $\Delta t[i]$ 中的高阶项。根据第二个假设，

$$q[n] = q[n-1] + q^{(1)}[n-1] \Delta t[n-1] + O(\Delta t[n-1]^2), \tag{35}$$

or

$$q^{(1)}[n-1] = \frac{q[n] - q[n-1]}{\Delta t[n-1]} + O(\Delta t[n-1]). \tag{36}$$

将其代入方程 34 的 $k = 1$项后，状态偏差的表达式简化为

$$\begin{aligned} \Delta x[n] &\simeq -\frac{1}{2} \sum_{i=0}^{n-1} (q[i+1] - q[i])\, \Delta t[i] \\ &= \frac{1}{2} q[0] \Delta t[0] + \frac{1}{2} \sum_{i=1}^{n-1} q[i] (\Delta t[i] - \Delta t[i-1]) - \frac{1}{2} q[n] \Delta t[n-1]. \end{aligned} \tag{37}$$

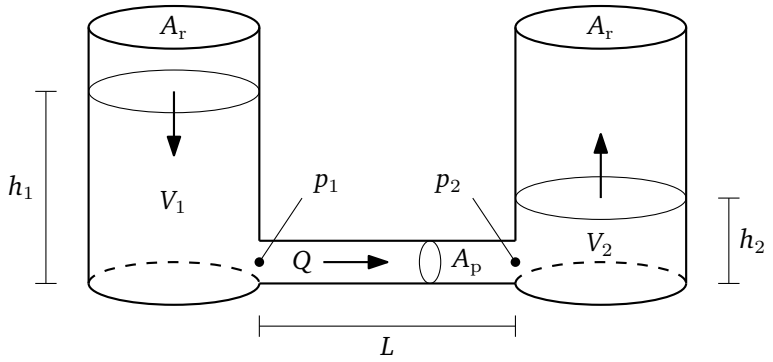

图7：相连流体储液罐示例的示意图。

这一般表明，步长变化会导致状态偏差，其主导项与相邻步之间的步长差以及中间通信点处的流量均成正比。

4 示例 2：相连流体储罐

有人可能会认为，我们在第2节中使用的阻尼谐振子示例过于理想，几乎像是专门设计来产生所展示的效果。状态 x_1，即从弹簧阻尼子系统的角度观察到的位移，可以说是多余的：将质量块的位移 x_2 视为“真实值”会更自然，而且很容易将其添加为 S_2 的输出，并将其作为 S_1 的输入。

因此，我们以另一个示例为本文作结，如图7所示，这是一个由管道连接的两个流体储罐模型。对于这个系统，将流变量用作两个仿真单元之间的接口是相当自然的，因为它在物理上的体现就是流体流动。这也是一个系统示例：虽然将冗余状态作为 S_2的输出非常简单，但对于各个子系统的建模者来说，这样做不太合理：在此情况下，流量积分就是从一个储液罐流入另一个储液罐的流体总量。然而，在仿真单元内部，对它的自然表示方式是各储液罐中的流体总体积。

我们将采用一种高度简化的系统表示方式，其中管道孔口处的压力就是储液罐底部的静水压力，

$$p_i = \rho g h_i = \frac{\rho g V_i}{A_\mathrm{r}}, \tag{38}$$

管道两端的压降由黏性牛顿流体层流的哈根–泊肃叶方程给出：

$$p_1 - p_2 = \frac{8\pi\mu L Q}{A_\mathrm{p}^2}. \tag{39}$$

从物理上来说，这对应于这样一种情况：相对于储液罐的尺寸，管道非常细长，流体缓慢地流过管道。在这些方程中， p_1 和 p_2 分别表示每个储液罐底部的压力，V_i 表示每个储液罐中的流体体积， Q 表示两个储液罐之间的体积流量。 A_r 表示储液罐的水平横截面积（假设相等）， A_p 表示管道的横截面积， L 表示管道的长度。ρ和 μ 分别表示流体的密度和动力黏度，而g 表示重力加速度。由于这些物理细节对我们的目的而言并不重要，为了使表达清晰，我们将符号简化为仅包含两个常数： C 和 R:

$$p_i = \frac{V_i}{C}, \qquad C \equiv \frac{A_\mathrm{r}}{\rho g}, \tag{40}$$

$$p_1 - p_2 = RQ, \qquad R \equiv \frac{8\pi\mu L}{A_\mathrm{p}^2}. \tag{41}$$

（这些符号分别类比于液压电容和液压电阻来选择。）

同样地，我们将系统拆分为两个仿真单元 S_1 和 S_2，前者表示储液罐1，后者表示储液罐2和管道。我们通过压力–流量耦合将它们连接起来，其中压力 p 是接口处的压力，即储液罐1底部的压力。设置如图8所示。 S_1 的状态空间方程为

$$\dot{V}_1(t) = -\tilde{u}_1(t), \tag{42}$$

$$y_1[n] = p_1[n] = \frac{V_1[n]}{C}, \tag{43}$$

而对于 S_2 ，方程为

$$\dot{V}_2(t) = Q(t) = \frac{\tilde{u}_2(t)}{R} - \frac{V_2(t)}{CR}, \tag{44}$$

$$y_2[n] = Q[n]. \tag{45}$$

注意， y_1 和 u_2 表示界面压力 p，而 u_1 和 y_2表示界面流量 Q。因此，耦合约束为

$$u_1[n] = y_2[n] \qquad \text{(flow rate)}, \tag{46}$$

$$u_2[n] = y_1[n] \qquad \text{(pressure)}. \tag{47}$$

与第一个示例一样，状态方程（42 和 44）有直接的解析解，因此我们不必担心数值积分引入的误差。

我们选择参数值 $C = 1$ 和 $R = 1$ ，以及初始条件 $V_1(0)$ =0.6 和 $V_2(0) = 0.4$。我们使用一种基于流量的“bang-bang”步长控制器：如果流量大于某个阈值（$Q_\mathrm{t} = 0.5$），则使用较小的步长（$\Delta t_\mathrm{small} = 0.001$）；否则，使用较大的步长（$\Delta t_\mathrm{large} = 0.01$）。也就是说，

$$\Delta t[n] = \begin{cases} \Delta t_\mathrm{small}, & y_2[n] > Q_\mathrm{t} \\ \Delta t_\mathrm{small}, & y_2[n] \le Q_\mathrm{t}. \end{cases} \tag{48}$$

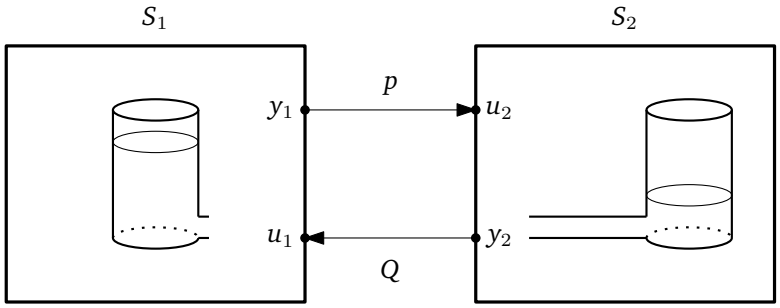

图1：相连流体储液罐示例的协同仿真设置。请注意， Q 为正表示流体从储液罐 1 流向储液罐 2，尽管我们选择的连接因果关系方向是从 S_2 到 S_1。 p 是两个子系统接口处的压力，也就是储液罐 1 底部的压力。

在 $t = 1$，我们（瞬间）将 S_1 中的流体体积增加 1，以增大流量并促使步长发生变化。

结果如图9所示。可以看到，两个储液罐之间初始的液位差呈指数衰减，且两者中的流体体积都趋近于 0.5 的平衡液位。在 $t = 1$，我们向储液罐 1 中加入单位体积的流体，此后液位最终在 $V_1 \approx V_2 \approx 1$处趋于一致。与第一个示例一样，在右侧放大图中可以看到固定步长仿真与变步长仿真之间的差异。图10突出显示了这一差异，我们再次看到，变步长情形下最终状态之间的偏差更大。这里，偏差计算为$\Delta V = V - V_1 - V_2$，其中 V 是加入系统的流体总体积（即， $t < 1$ 时为 $V = 1$， $t \geq 1$时为 $V = 2$）。

5 结论

在协同仿真中，若两个仿真单元通过某个变量q相连，且两个仿真单元都具有表示 q时间积分的内部状态，那么由于外推误差，这些状态之间通常会存在偏差。若整体减小宏观时间步长，这类外推误差就会减小，因为一个仿真单元中对 q 的重构将趋近于另一仿真单元所定义的 q 的真实函数形式。反之，若整体增大步长，则外推误差可能会增大。

然而，本文表明，当步长随时间以非均匀方式变化时，情况会更加复杂。在通信点 n 处，步长从 Δt 变为 $\Delta t\prime$，这会对积分状态之间累积的偏差产生一个主导阶贡献，其大小与 $q[n](\Delta t\prime - \Delta t)$成正比。

这主要是变步长仿真的一个问题，因为其中 q会随时间变化多个数量级。要理解其原因，不妨再次考虑

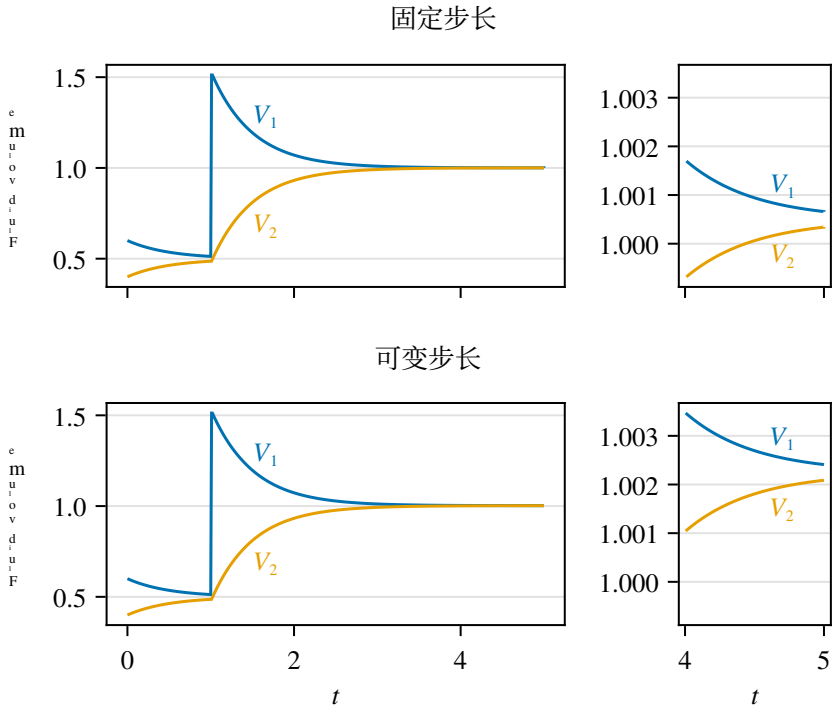

图 9：连接流体储液罐示例的协同仿真结果。上方各图显示采用固定步长得到的解；下方各图则对应变步长算法。最左侧各图显示整个仿真过程；最右侧各图展示了尾部的放大图（$4 \le t \le 5$）。

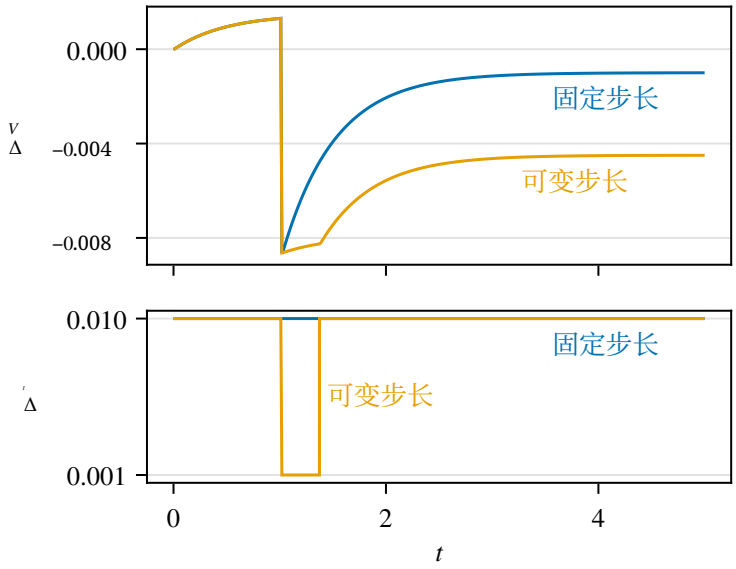

图10：上图显示了在固定步长和变步长情形下， S_1 与 S_2、 $\Delta V = V - V_1 - V_2$之间位移状态的偏差。下图显示了宏观时间步长。

状态偏差的一般渐近表达式（37）：

$$\begin{aligned}\Delta x[n] &\simeq -\frac{1}{2}\sum_{i=0}^{n-1}\left(q[i+1]-q[i]\right)\Delta t[i] \\ &= \frac{1}{2}q[0]\Delta t[0] + \frac{1}{2}\sum_{i=1}^{n-1} q[i]\left(\Delta t[i]-\Delta t[i-1]\right) - \frac{1}{2}q[n]\Delta t[n-1].\end{aligned}$$

第一行显然表明，如果 q 变化不大，即对于所有 i， $q[i+1] \approx q[i]$都成立，那么累积偏差就会很小。根据第二行的解释，这是因为步长随时间增大和减小的变化会相互抵消。另一方面，如果 q 变化很大，并且在 q 的绝对值较大时恰好发生了较大的步长变化，就可能造成无法保证会在之后抵消的偏差。这正是我们在两个示例中观察到的现象。

为避免此类错误，可以限制步长控制器在其引起的步长变化幅度方面的激进程度。对于我们在第2节中使用的PI控制器，这意味着使参数 $\theta_{\min}$和 $\theta_{\max}$ 更接近1，或使用较低的PI增益。[2] 对于初始阶段动态性很强、但最终会达到稳态的仿真，从最小步长开始运行可变步长控制器可能会有所帮助。这是

[2]在那个特定示例中，明显的罪魁祸首是参数 $\theta_{\min} = 0。2$，其值直接取自原始论文 [8], ，该论文允许在单次跳变中将步长最多缩短80%。

使用反馈步长控制器时的一般建议 [11],，在这种情况下尤其如此。最后，可以设想开发一种监测流量型变量的步长控制器，并在步长变化较大时对其加以限制。协同仿真接口可以通过支持为变量添加值范围和物理单位注释来促进这一点，例如功能样机接口（FMI） [13]。

最后，我们指出本研究的主要局限：我们只考虑了常量输入外推（零阶保持，ZOH），并且采用了显式雅可比协同仿真算法。这两者在各自方面都属于“最简单”的选择，其优点是我们能够对这一问题进行清晰、直接的分析。由于观察到的偏差是由外推误差造成的，因此可以预期，高阶外推以及采用隐式协同仿真或高斯–赛德尔迭代会减弱这种影响，甚至可能使其在 Δt 中提升到更高阶。这或许是未来研究的一个有趣方向。

可复现性

本文用于运行仿真和生成图表的代码可在线获取：https://doi.org/10.60609/6p8m-0713。代码使用 Julia 编程语言 [14] 编写，并使用 Makie.jl [15] 生成图表。

致谢

本研究得到挪威研究理事会资助，资助项目包括*SEACo*：更安全、更简便且更准确的协同仿真（资助编号 326710）和*OptiStress*：协同仿真中的系统优化与压力测试（资助编号 344238）。

作者感谢同事 Severin Sadjina 和 StianSkjong 就本文所 ներկայաց的工作展开了富有意义的讨论并提供了反馈。

参考资料

[1] Cláudio Gomes 等。《协同仿真：综述》。载于：*ACM*计算调查 51.3，49（2018 年 5 月），第 1–33 页。doi:10.1145/3179993。[2] R. Kübler 和 W. Schiehlen。《模拟器耦合的两种方法》。载于：*Mathematical and Computer Modellingof Dynamical Systems*6.2（2000 年），第 93–113 页。doi: 10.1076/1387-3954(200006)6:2;1-M;FT093。[3] Martin Arnold 和 Michael Günther。《耦合微分代数系统的预条件动态迭代》。载于：*BIT* 数值数学41.1（2001 年 1 月），第 1–25 页。doi: 10.1023/A:1021909032551。[4] Martin Arnold、Christoph Clauss 和 Tom Schierz。《用于 FMI 模型交换和协同仿真 V2.0 的协同仿真误差分析与误差估计》。载于：机械工程档案60.1（2013 年），第 75–94 页。doi: 10.2478/meceng-2013-0005。

[5] Slaven Glumac 和 Zdenko Kovačić。《非迭代协同仿真的缺陷分析》。载于：*Mathematics*11.6，1342（2023 年）。doi: 10.3390/math11061342。[6] Martin Busch 和 Bernhard Schweizer。《控制协同仿真方法宏步长的显式方法》。载于：第七届欧洲非线性动力学会议论文集。ENOC 2011（意大利罗马，2011 年 7 月 24–29 日）。Davide Bernardini、Giuseppe Rega 和 Francesco Romeo 编。2011 年。isbn: 9788890623424。[7] Tom Schierz、Martin Arnold 和 Christoph Clauß。《采用通信步长控制的 FMI 兼容主算法进行协同仿真》。载于：第九届国际 *MODELICA* 会议论文集（德国慕尼黑，2012 年 9 月 3–5 日）。Martin Otter 和 Dirk Zimmer 编。林雪平电子会议论文集 76。2012 年 11 月 19 日，第 205–214 页。isbn: 978-91-7519-826-2。doi: 10.3384/ecp12076205。[8] Severin Sadjina 等。《协同仿真中的能量守恒与功率键：非迭代自适应步长控制和误差估计》。载于：*Engineering with Computers* 33（2017 年），第 607–620 页。doi: 10.1007/s00366-016-0492-8。[9] T. Meyer、J. Kraft 和 B. Schweizer。《协同仿真：误差估计与宏步长控制》。载于：*JournalofComputational and NonlinearDynamics* 16.4，041002（2021 年 4 月）。doi: 10.1115/1.4048944。[10] Yohan Éguillon、Bruno Lacabanne 和 Damien Tromeur-Dervout。《F_3ORNITS：一种灵活的可变步长非迭代协同仿真方法，用于处理具有混合高级功能的子系统》。载于：*Engineering with Computers*38（2022 年 8 月），第 4501–4543 页。doi: 10.1007/s00366-022-01610-z。[11] Lars T. Kyllingstad、Severin Sadjina 和 Stian Skjong。《采用最小子系统接口的误差估计与步长控制》。载于：*SimulationModellingPracticeand Theory* 145，103209（2025 年 12 月）。doi: 10.1016/j.simpat.2025.103209。[12] Stian Skjong 和 Eilif Pedersen。《动态分布式仿真中的数值稳定性》。载于：*Mathematics andComputers in Simulation* 163（2019 年 9 月），第 183–203 页。doi: 10.1016/j.matcom.2019.02.018。[13] The Modelica Association。《功能模型接口规范》。版本 3.0.2。2024 年 11 月 27 日。url: https://fmi-standard.org（访问日期：2025 年 11 月 12 日）。[14] Jeff Bezanson 等。《Julia：一种数值计算的新方法》。载于：*SIAM Review* 59.1（2017 年），第 65–98 页。doi: 10.1137/141000671。[15] Simon Danisch 和 Julius Krumbiegel。《Makie.jl：用于 Julia 的灵活高性能数据可视化》。载于：*Journal of Open SourceSoftware*6.65，3349（2021 年 9 月 1 日）。doi: 10.21105/joss.03349。