

DOI:10.19853/j.zgjsps.1000-4602.2020.17.018

# 基于 CUDA 架构的内涝一维/二维耦合模型求解方法

金 溪<sup>1</sup>, 王 芳<sup>2</sup>

(1. 武汉理工大学 土木工程与建筑学院, 湖北 武汉 430070; 2. 武汉市规划研究院, 湖北 武汉 430010)

**摘 要:** 以二维浅水方程为基础建立二维水动力模型, 并利用二阶 Godunov 格式有限体积法进行求解, 从而获得地表二维水动力模型的求解模块。将其与 SWMM5.1 一维水动力模型进行耦合, 实现一维、二维水动力模型的耦合模拟。为了提高耦合模型的求解效率, 将 CUDA 并行计算架构引入耦合模型求解过程中。将一维水动力模型的求解过程在主机端运行, 将二维水动力模型的求解过程在设备端运行。利用 cudaMemcpy 函数实现一维、二维模型之间地面水深数据交换, 并利用地面水深数据进行一维、二维模型之间双向流量数值的计算, 从而实现一维、二维模型的耦合。通过实际案例验证表明, CUDA 并行计算架构的引入可以在计算效率方面较非并行求解方法提升 6~8 倍, 计算误差方面可以保证水量平衡相对误差小于 0.1%, 积水深度误差小于 0.01 m。该方法可以用于城市内涝风险评估及排水系统规划等场合。

**关键词:** CUDA; 浅水方程; 水动力模型; 耦合模拟; 城市内涝

**中图分类号:** TU992 **文献标识码:** A **文章编号:** 1000-4602(2020)17-0103-07

## Calculation Method of Urban Flooding One-dimensional/Two-dimensional Coupling Model Based on CUDA-based Parallel Implementation

JIN Xi<sup>1</sup>, WANG Fang<sup>2</sup>

(1. School of Civil Engineering and Architecture, Wuhan University of Technology, Wuhan 430070, China; 2. Wuhan Planning and Design Institute, Wuhan 430010, China)

**Abstract:** A two-dimensional (2D) hydrodynamic model was built based on the shallow water equation. The equation was then solved by using a second order Godunov-type scheme, and the solution module of the 2D surface hydrodynamic model was obtained. The 2D hydrodynamic module was coupled with SWMM5.1 one-dimensional (1D) hydrodynamic model, and coupling simulation of 1D and 2D hydrodynamic models was realized. In order to improve the solution efficiency of the coupling model, CUDA-based parallel implementation was introduced into the solution process. Solution process of the 1D hydrodynamic module was ran on the host side, and the 2D hydrodynamic module was ran on the device side. Exchange of surface water depth data between 1D and 2D models was realized by using cudaMemcpy function, and water volume exchange between 1D and 2D models was calculated according to the surface water depth, so that coupling of the 1D and 2D models was achieved. Tested by a real study case, introduction of CUDA-based parallel implementation improved the solution efficiency by 6 to 8 times compared with non-parallel solution, and the relative error of water balance was less than 0.1%

and the error of water depth was less than 0.01 m. Therefore, the calculation method can be used in cases such as risk assessment of urban flooding and storm water drainage system planning.

**Key words:** CUDA; shallow water equation; hydrodynamic model; coupling simulation; urban flooding

在城市内涝二维水动力模型中,地面水流的流动状态一般采用浅水方程组(SWEs)进行描述。近年来基于 Godunov 格式的有限体积法逐渐成为求解 SWEs 的主流方法<sup>[1-4]</sup>,并且被广泛应用于城市内涝数值模型中。但是由于二维水动力模型求解的复杂性,在大范围、长历时的内涝模拟过程中,计算效率成为制约该项技术使用的一个障碍。围绕提高模型求解效率许多研究者展开了大量研究,例如通过单元格移除法<sup>[5]</sup>、局部时间步长法<sup>[6]</sup>、双时间步长法<sup>[7]</sup>等方法提高计算效率,其中单元格移除法可以在保证计算精度的前提下减少计算耗时,但是其计算效率提高的幅度与地面积水面积大小成反比,当地面积水较多时计算效率下降明显。局部时间步长法及双时间步长法可以较大幅度地提高计算效率,但是这两种方法的使用会导致计算精度退化。

近年来随着并行计算方法的兴起,为二维水动力模型的快速求解提供了一条新的路径。但是由于 CPU 的物理计算核心的数量限制,CPU 并行计算技术从微观角度来说,多个线程在一定程度上仍然是串行的,因此效率提升有限。与 CPU 相比,GPU 具有更多的计算核心(可达上千个),虽然单核计算能力较 CPU 弱,但是非常适合处理数据量大、逻辑关系简单的数值计算。对于数据耦合性弱且逻辑简单的模拟问题,GPU 并行计算的效率可达到 CPU 计算效率的几十乃至上百倍。由于二维水动力模型采用空间离散化框架进行求解,并且数据量大,因此非常适用于 GPU 并行计算方法。Liang 等人<sup>[8]</sup>采用 Open CL 技术利用 GPU 对计算过程进行加速,许栋等人<sup>[9]</sup>采用 CUDA 架构实现了对城市洪水演进的高效模拟,均取得了较好的效果。但是上述研究只针对单纯的二维水动力模型的求解,没有将城市排水管网的作用考虑在内。而城市内涝的积水、退水过程中,城市排水管网的作用是不能忽略的,为准确模拟城市内涝的积水、退水过程,需要将城市排水管网的一维水动力模型与地表二维水动力模型进行耦合求解。针对这一问题,笔者将采用 CUDA 框架给出基于并行计算的一维、二维水动力耦合模型的高

效求解方案。

## 1 二维浅水控制方程

二维守恒型浅水方程可写成如下的矩阵形式:

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = S \quad (1)$$

式中: $U$ 、 $F$ 、 $G$ 、 $S$  分别表示守恒向量、 $x$  和  $y$  方向的通量向量以及源项向量, $t$ 、 $x$ 、 $y$  分别代表时间以及直角坐标系的坐标轴。 $U$ 、 $F$ 、 $G$ 、 $S$  的具体形式如下:

$$U = \begin{bmatrix} \eta \\ uh \\ vh \end{bmatrix}, F = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}(g\eta^2 - 2\eta z_b) \\ huv \end{bmatrix}$$

$$G = \begin{bmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}(g\eta^2 - 2\eta z_b) \end{bmatrix}, S = R + S_f + S_b \quad (2)$$

式中: $\eta$  表示水面的水位; $u$ 、 $v$  表示  $x$  与  $y$  方向的流速; $h$  表示水深; $g$  为重力加速度; $z_b$  表示地面绝对高程; $R$  表示一维模型溢流源项; $S_b$  表示地表坡度源项, $S_f$  表示地面摩阻源项。

## 2 二维水动力模型数值求解方法

具有数值和谐性的式(1)和式(2)采用二阶 Godunov 格式的有限体积法进行求解。利用 MUSCL 线性重构方法及 MINMOD 坡度限制器求解单元格边界处守恒变量的插值结果,利用该插值结果进行单元格的界面通量及时间推进公式的计算,可以获得空间上具有二阶精度的计算结果。时间上的二阶精度通过二阶 Runge-Kutta 方法对时间项进行离散来获得,时间项的离散如下所示:

$$U_{i,j}^{n+1} = U_{i,j}^n + \frac{1}{2}\Delta t [K_{i,j}(U^n) + K_{i,j}(U^*)] \quad (3)$$

为了保证线性重构过程中水深为非负值,采用非负水深重构方法对 MUSCL 线性重构方法获得的界面重构值及坡度源项进行修正。算法流程及修正过程可参考文献[10]。利用修正后的界面重构值及坡度源项进行界面通量及时间推进公式的求解。

### 3 一维/二维耦合模拟策略及并行实现

#### 3.1 一维/二维模型的时间同步

耦合模拟过程中,一维模型及二维模型在每个一维模型时间步长的结束时刻进行时间的同步以及流量的交换。利用式(4)计算二维模型的时间步长,从而实现一维/二维模型的时间同步。

$$\Delta t_{2D} = \min(\Delta t_{CFL}, t_{1D} - t_{2D}) \quad (4)$$

式中: $\Delta t_{2D}$ 为二维模型的时间步长; $\Delta t_{CFL}$ 为根据CFL准则计算出的时间步长; $t_{1D}$ 为一维模型当前步长的结束时刻; $t_{2D}$ 为二维模型上一个步长的结束时刻。一维/二维模型时间步长的关系以及数据交换的时刻如图1所示,其中, $\Delta t_{1D}$ 为一维模型的时间步长, $Q_{1D \rightarrow 2D}$ 为一维模型溢流至二维模型的节点溢流体积, $Q_{2D \rightarrow 1D}$ 为二维模型回流至一维模型的体积。

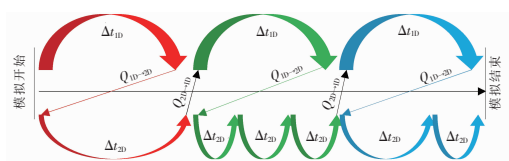


图1 一维/二维模型的时间步长关系

Fig. 1 Relationship of 1D/2D models time steps

从图1可以看出,时间步长的同步是通过调整二维模型的时间步长来实现的,当通过CFL准则计算获得的二维模型时间步长大于一维模型时间步长时,将二维模型时间步长设置为与一维模型时间步长相等。否则,一维模型的一个时间步长中会包含多个二维模型的时间步长,并且强制最后一个二维模型的时间步长结束时刻与其对应的一维模型时间步长的结束时刻一致。

#### 3.2 一维/二维模型间水量交换机制

在每个一维模型时间步长模拟结束的时刻,将一维模型中溢流节点的溢流体积作为溢流源项赋给溢流节点所在的二维模型地表面单元格,如图1中 $Q_{1D \rightarrow 2D}$ 所示,在本研究中采用SWMM中节点的溢流流量(flooding)计算结果与时间步长 $\Delta t_{1D}$ 的乘积作为 $Q_{1D \rightarrow 2D}$ 的值。 $Q_{1D \rightarrow 2D}$ 的值将作为一维模型溢流源项 $R$ 参与二维模型计算。

在每个一维模型时间步长内的最后一个二维模型时间步长结束时刻,遍历一维模型所有节点并计算二维模型回流至一维模型节点的回流体积 $Q_{2D \rightarrow 1D}$ ,在计算回流体积之前需要先判断节点是否具有回流条件,判断依据为该节点在一维模型当前时间步长 $\Delta t_{1D}$ 中的水头 $H_{1D}$ 是否小于节点所在二维

模型单元格的水位 $H_{2D}$ ,若 $H_{2D} > H_{1D}$ 则计算回流体积,并且回流的作用水头 $\Delta H = H_{2D} - H_{1D}$ 。节点回流体积采用如下公式计算:

$$Q_{2D \rightarrow 1D} = \sum_{i=0}^n (Q_i \cdot \Delta t_i) \quad (5)$$

式中: $n$ 为当前一维模型时间步长中对应的二维模型时间步长数量; $\Delta t_i$ 为一维模型时间步长中第 $i$ 个二维模型的时间步长; $Q_i$ 为 $\Delta t_i$ 时间步长中的回流流量, $Q_i$ 采用堰流公式(6)<sup>[11]</sup>计算。

$$Q_i = c_w \cdot \omega \cdot \Delta H \sqrt{2g\Delta H} \quad (6)$$

式中: $c_w$ 为堰流系数; $\omega$ 为节点过流宽度。 $Q_{2D \rightarrow 1D}$ 将作为一维模型节点的进流量(inflow)参与一维模型的计算。通过 $Q_{1D \rightarrow 2D}$ 、 $Q_{2D \rightarrow 1D}$ 的求解并参与到相应的一维、二维模型的求解过程中实现了一维、二维模型的耦合模拟。

#### 3.3 一维/二维耦合模拟的并行实现

耦合模拟过程中,二维浅水方程求解占用了总求解时间的绝大部分,因此这里仅将二维水动力模型的求解放在设备端(GPU),利用CUDA架构实现并行计算,一维管网模型的求解采用SWMM5.1模块在主机端(CPU)以串行方式计算。耦合模拟流程如图2所示。

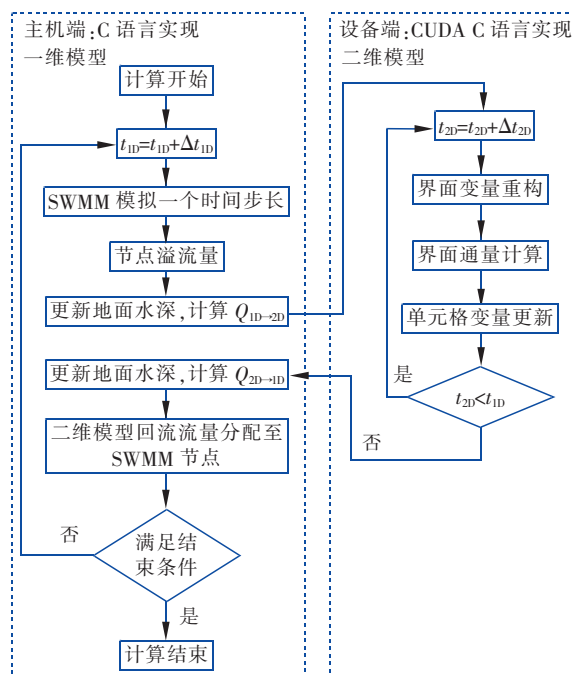


图2 一维/二维耦合模拟流程

Fig. 2 Flow chart of 1D/2D coupled modelling

由于一维模型在主机端运行,二维模型在设备端运行,主机端和设备端各自拥有独立的物理内存,

必须调用 `cudaMemcpy` 函数才能实现数据交互,而数据交互速度受带宽限制,往往成为 CUDA 并行计算效率提高的瓶颈。为了提高计算效率,仅在主机端与设备端之间进行地表积水深度数据的交换, $Q_{1D \rightarrow 2D}$ 、 $Q_{2D \rightarrow 1D}$  的求解以及其对地表水深的影响均在主机端计算完成。

二维模型地表网格采用结构化网格,由于 CUDA 架构采用二维网格形式组织计算线程,因此结构化网格更加容易实现地表网格与线程网格之间的一一对应。线程网格中  $x$ 、 $y$  方向线程块的数量利用如下公式计算:

$$b_x = \text{INT}\left(\frac{n_x + th_x}{th_x}\right); b_y = \text{INT}\left(\frac{n_y + th_y}{th_y}\right) \quad (7)$$

式中: $b_x$ 、 $b_y$  为  $x$ 、 $y$  方向的线程块数量; $th_x$ 、 $th_y$  为线程块中  $x$ 、 $y$  方向的线程数量,由于 GPU 中是以线程束(wrap)为基本单位进行并行计算,线程束中的线程数量为 32,为了尽可能充分利用 GPU 线程并发性能,应保证线程块中的线程数量为 32 的倍数,因此规定  $th_x$ 、 $th_y$  数值为 16 (线程块的线程数为 256); $n_x$ 、 $n_y$  为地面结构化网格  $x$ 、 $y$  方向的网格数量;INT 为取整操作。

#### 4 案例分析

WC 区位于武汉,将其雨水管网模型作为耦合计算的一维模型。WC 区的地形及雨水管网结构见图 3。二维计算区域长 17.78 km、宽 13.34 km,结构化网格尺寸为 20 m × 20 m,网格数量约为 60 万。

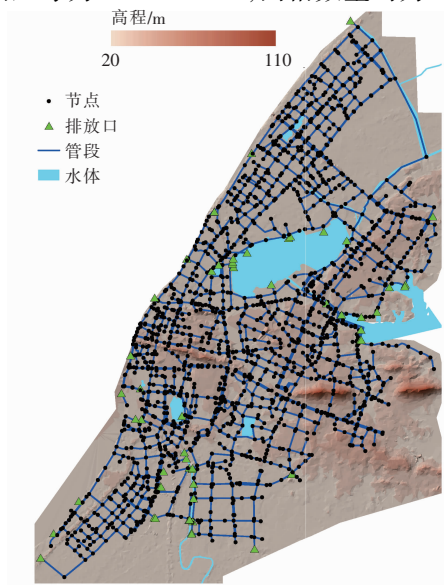


图3 研究区域的地形及管网结构

Fig. 3 Topography and drainage pipe network of study area

利用 CUDA 并行求解方法对该区域不同重现期降雨下的内涝情景进行模拟。各个重现期的降雨强度过程线如图 4 所示。

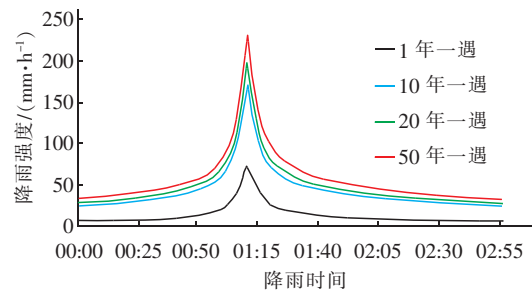


图4 不同重现期的降雨强度过程线

Fig. 4 Rainfall intensity curve with different return periods

作为比较,两种非并行耦合求解方法(普通耦合合法、单元格移除法)也被用来对上述情景进行模拟,这两种方法的一维/二维耦合机制与并行方法一致,但一维、二维模拟均在主机端采用非并行方法进行,普通耦合合法在二维模型求解过程中所有单元格不考虑干湿状态均参与计算。单元格移除法中,在二维模型求解过程中干单元格不参与计算。通过上述 3 种方法的对比,可以从模拟结果及计算效率方面对 CPU 端非并行模拟及 GPU 端并行模拟进行全面的比较,除了可以比较 GPU 并行算法与普通耦合方法在计算效率方面的差异以外,还可以与 CPU 非并行条件下效率较高的单元格移除方法的效率进行比较。

本研究中使用的主机端型号为 Intel Core i7 - 7700HQ,内存为 8 GB;设备端型号为 Nvidia GeForce GTX 1050,显存为 2 GB,流处理器数量为 640,流多处理器数量为 5,块最大线程数量为 1 024,网格最大线程数量为 2 147 483 647。

#### 5 结果与讨论

耦合模拟过程中,一维、二维模型之间的水量平衡必须得到满足,即在模拟的每个一维模型的步长中必须满足如下公式:

$$\Delta Q = Q_{1D \rightarrow 2D} - Q_{2D \rightarrow 1D} \quad (8)$$

式中: $\Delta Q$  表示一个时间步长内地面积水的变化量。

图 5 展示了 50 年一遇降雨条件下一维、二维模型之间的水量平衡计算结果,以及相对应的地表总积水量的变化过程。为了便于观察,将  $Q_{2D \rightarrow 1D}$  的值转化为负数后绘制于图中。



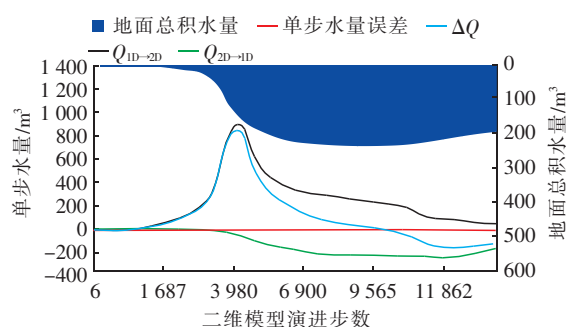


图 5 水量平衡误差及地面总积水量时间过程线

Fig. 5 Time duration curves of mass balance error and ponding volume

两种非并行耦合求解方法的计算结果完全一致,并且与 CUDA 方法的计算结果非常接近,图 5 根据 CUDA 方法的结果绘制而成。可以看出,  $\Delta Q$ 、 $Q_{1D \rightarrow 2D}$ 、 $Q_{2D \rightarrow 1D}$  很好地满足了水量平衡的条件,模拟结束时刻地面积水量为  $1\,983\,897\text{ m}^3$ ,水量平衡总误差为  $1\,231\text{ m}^3$ ,相对误差仅为  $0.062\%$ 。

由于 CPU 与 GPU 在浮点数的存储位数及截断方式上存在一定的差异<sup>[12]</sup>,因此 CPU 与 GPU 在进行浮点数运算过程中计算结果会存在微小的差异,但是对于二维水动力模拟来说,这种微小的差异可能会在计算过程中逐渐累积,从而导致最终结果出现明显的差异。为了降低这种差异导致的影响,所有计算数据在设备端及主机端均采用精度最高的双精度格式进行存储和计算。图 6 展示了不同降雨重现期条件下, CUDA 并行求解方法与非并行求解方法之间地面积水量的相对误差。

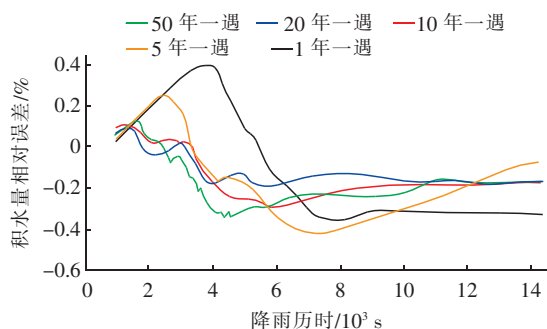


图 6 不同算法积水量相对误差的时间过程线

Fig. 6 Time duration curves of relative error of ponding volume between different algorithms

图 7 展示了 50 年一遇降雨条件下模拟结束时刻的积水深度及其误差情况,左侧 a 图为 CUDA 算法获得的积水深度,右侧 b 图为非并行算法与

CUDA 算法得到的积水深度的差值。通过图 6、图 7 可以看出,积水量误差以及积水深度误差均非常小,说明在双精度格式下 CUDA 方法的计算结果与非并行方法的计算结果差异很小,其计算结果是可以接受的。

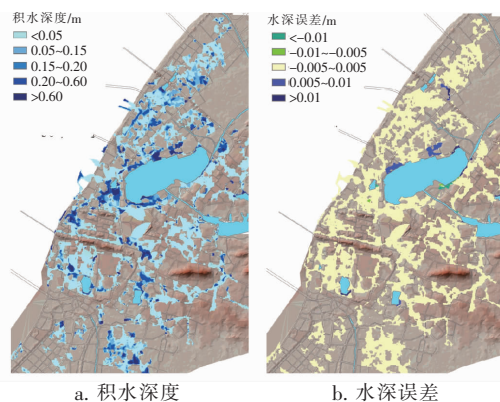


图 7 积水深度及积水深度误差色阶图

Fig. 7 Colour charts of ponding depth and its error

3 种计算方法的二维水动力模型单步计算耗时随演进步数的过程线如图 8 所示。可以看出,对于非并行算法,采用单元格移除方法时由于干单元格不参与计算,大大减少了二维模型的计算量,从而显著地提高了二维水动力模型的求解效率。尤其是在降雨强度较小(1 年一遇)的情况下,其单步计算耗时在整个计算过程中甚至始终小于 CUDA 并行求解方法。但是其计算效率的高低与地面积水范围的大小密切相关,随着积水面积的扩大,参与计算的单元格数量也随之增多,导致单步计算耗时显著增加,在 10、20、50 年一遇降雨条件下,单步计算耗时随着积水面积的增大而迅速上升,并超过 CUDA 并行计算方法的单步计算耗时。CUDA 并行计算方法的单步计算耗时较非并行-普通耦合计算方法有极大的提高,并且计算效率稳定,与地面积水情况无关,在多数情况下其计算效率较非并行-单元格移除方法的计算效率更高。但是 CUDA 方法计算效率的提高程度较参考文献[8-9]中所展示的计算效率提高程度要低。这主要是因为文献[8-9]中仅仅针对二维水动力模型进行求解,因此不需要频繁地进行设备端与主机端的数据交换,而在一维、二维耦合模拟过程中,为了进行一维模型回流量及地表溢流源项的更新,需要频繁地进行主机端与设备端的数据交换,从而导致计算效率的提高程度不如单纯二维水动力模型并行求解计算效率的提高程度。

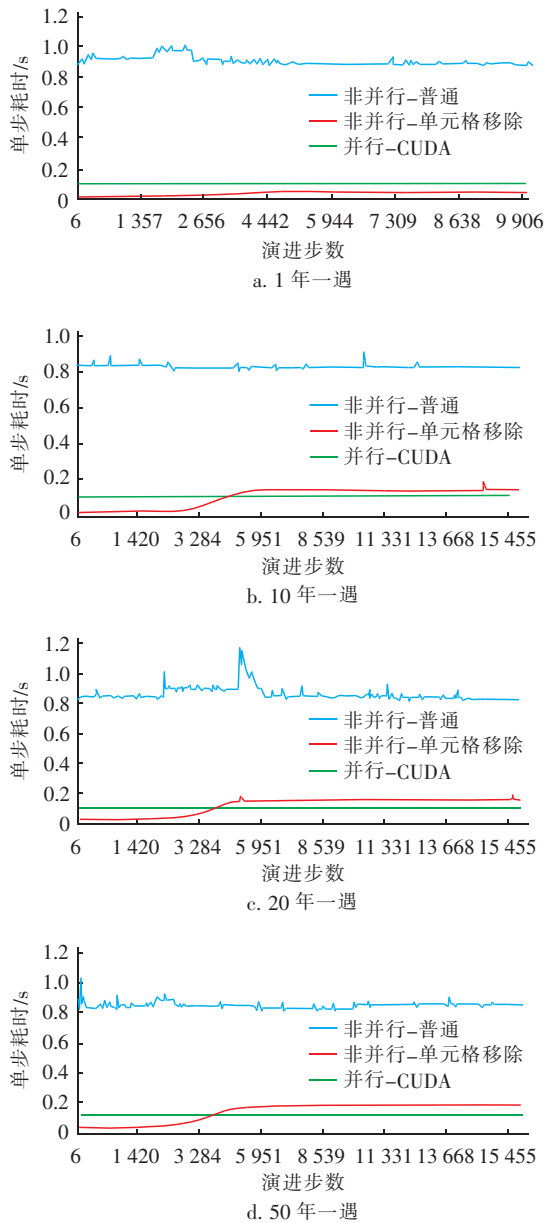


图8 不同求解方法单步计算耗时随演进步数的过程线

Fig.8 Duration curves of computational time consumption per step of different solving methods

3种计算方法在不同降雨条件下的总计算耗时见表1。总体而言,CUDA并行求解方法是计算效率最高的,只有在地面积水较少的场合,非并行-单元格移除法的计算效率才比CUDA并行求解方法的计算效率要高。

通过上述分析可知,CUDA架构下的一维、二维水动力模型耦合计算方法在水量平衡、计算精度及求解效率方面均具有较好的表现,应用在城市内涝分析中是可行的。

表1 不同计算方法的总计算耗时比较

Tab.1 Computational time consumption of different solving methods

重现期/年	计算方法	耗时/s	加速比
1	非并行-普通	9 350	1.00
	非并行-单元格移除	463	20.19
	并行-CUDA	1 170	7.99
10	非并行-普通	13 264	1.00
	非并行-单元格移除	1 919	6.91
	并行-CUDA	1 827	7.26
20	非并行-普通	13 726	1.00
	非并行-单元格移除	2 048	6.70
	并行-CUDA	1 830	7.50
50	非并行-普通	14 105	1.00
	非并行-单元格移除	2 373	5.94
	并行-CUDA	1 905	7.40

## 6 结论

① 以二维浅水方程为基础建立二维水动力模型,并利用非负水深重构方法为基础的二阶Godunov格式有限体积法进行求解,从而获得地表二维水动力模型的求解模块。将其与SWMM5.1一维水动力模型进行耦合,实现一维、二维水动力模型的耦合模拟。在耦合模拟过程中,通过调整二维水动力模型的时间步长,在每个一维模型时间步长的结束时刻实现一维、二维耦合模拟的时间同步及一维、二维模型间的水量交换。实际案例计算结果表明,该耦合方法可以较好地模拟城市降雨导致的内涝情景,能够在考虑城市排水管网排水能力的条件下给出地面积水随时间的变化情况。

② 为了提高耦合模型的求解效率,将CUDA并行计算架构引入耦合模型求解过程中。将一维水动力模型的求解过程在主机端运行,将二维水动力模型的求解过程在设备端运行。利用cudaMemcpy函数实现一维、二维模型之间的地面水深数据交换,并利用地面水深数据进行一维、二维模型间双向流量数值的计算,从而实现一维、二维模型的耦合。实际案例计算结果表明,CUDA并行计算架构的引入可以大大提高二维水动力模型的求解效率,相较于非并行-普通耦合模拟方法计算效率可以提升6~8倍,相较于非并行-单元格移除耦合模拟方法计算效率也有5%~20%的提升。在采用双精度数据格式的情况下,CUDA并行计算结果与非并行计算结果之间水量平衡的相对误差小于0.1%、积水深度的误差小于0.01 m,完全可以用于城市内涝风险

评估及排水系统规划等场合。但是由于耦合模拟过程中需要频繁进行一维、二维模型之间的数据交换,导致 CUDA 并行架构计算效率的提高幅度不如单纯求解二维水动力模型时明显。

#### 参考文献:

- [1] Xia X, Liang Q. A new efficient implicit scheme for discretising the stiff friction terms in the shallow water equations[J]. Adv Water Resour, 2018, 117: 87–97.
- [2] Delis A I, Nikolos I K. A novel multidimensional solution reconstruction and edge-based limiting procedure for unstructured cell-centered finite volumes with application to shallow water dynamics [J]. International Journal for Numerical Methods in Fluids, 2013, 71(5): 584–633.
- [3] Hou J, Simons F, Mahgoub M, *et al.* A robust well-balanced model on unstructured grids for shallow water flows with wetting and drying over complex topography [J]. Computer Methods in Applied Mechanics and Engineering, 2013, 257: 126–149.
- [4] Li G, Song L, Gao J. High order well-balanced discontinuous Galerkin methods based on hydrostatic reconstruction for shallow water equations[J]. Journal of Computational and Applied Mathematics, 2018, 340: 546–560.
- [5] Falconer R A, Owens P H. Numerical simulation of flooding and drying in a depth-averaged tidal flow model [J]. Proceedings of the Institution of Civil Engineers, 1987, 83(1): 161–180.
- [6] Sanders B F. Integration of a shallow water model with a local time step[J]. J Hydraul Res, 2008, 46(4): 466–475.
- [7] Yu H, Huang G, Wu C. Efficient finite-volume model for shallow-water flows using an implicit dual time-stepping method[J]. J Hydraul Eng, 2015, 141(6): 04015004.
- [8] Liang Q, Smith L S. A high-performance integrated hydrodynamic modelling system for urban flood inundation[J]. Journal of Hydroinformatics, 2015, 17(4): 518–533.
- [9] 许栋, 徐彬, David P Ayet, 等. 基于 GPU 并行计算的浅水波运动数值模拟[J]. 计算力学学报, 2016, 33(1): 113–120.
- Xu Dong, Xu Bin, David P Ayet, *et al.* Numerical simulation of shallow water motion based on parallel computation using GPU [J]. Chinese Journal of Computational Mechanics, 2016, 33(1): 113–120 (in Chinese).
- [10] Liang Q. Flood simulation using a well-balanced shallow flow model[J]. J Hydraul Eng, 2010, 136(9): 669–675.
- [11] Leandro J, Martins R. A methodology for linking 2D overland flow models with the sewer network model SWMM5.1 based on dynamic link libraries[J]. Water Sci Technol, 2016, 73(12): 3017–3026.
- [12] Whitehead N, Fit-Florea A. Precision & Performance: Floating Point and IEEE 754 Compliance for NVIDIA GPUs[M]. Santa Clara: NVIDIA, 2011.



作者简介:金溪(1978–),男,河北河间人,博士,讲师,主要研究方向为城市给排水系统优化与设计。

E-mail: jinxixi@whut.edu.cn

收稿日期:2019–06–27