Computer Science and Application 计算机科学与应用, 2021, 11(12), 2845-2852
Published Online December 2021 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2021.1112289

Hans汉斯

基于OpenMP的明渠圣维南系统的大规模模拟

齐永孟

青岛大学，山东 青岛

收稿日期：2021年11月3日；录用日期：2021年12月1日；发布日期：2021年12月8日

摘　要

了解流道中水流的行为对于早期洪水灾难管理和挽救生命起到了至关重要的作用。本文就是以地表水流作为研究对象，了解洪水的行为。为了预测和模拟洪水的演进过程，利用二维圣维南偏微分方程建立了具有初始条件和边界条件的数学模型。使用显示的有限差分法对模型进行离散化，在时间和空间上均使用中心差分格式，时间上的中心差分也被称作蛙跳格式。之后使用OpenMP对其并行化实现。为了测试和实验的目的，我们使用了一个简单的长方体流道来模拟。通过数值模拟得到不同时间步长下的输出参数，比如水流的高度、速度，之后对这些参数进行处理，实现可视化。最后，将并行程序与串行程序进行对比，进行规模扩展测试。

关键词

圣维南方程，浅水方程，有限差分法，跳蛙格式，并行计算，OpenMP

Large-Scale Simulation of the Open Channel Saint-Venant System Based on OpenMP

Yongmeng Qi

Qingdao University, Qingdao Shandong

Received: Nov. 3rd, 2021; accepted: Dec. 1st, 2021; published: Dec. 8th, 2021

Abstract

Understanding the flow behavior in the channel plays an important role in early flood disaster management and lifesaving. This paper is aimed at the surface flows to study the behavior of flood waves. In order to predict and simulate the flood process, a mathematical model with initial and

文章引用: 齐永孟. 基于 OpenMP 的明渠圣维南系统的大规模模拟[J]. 计算机科学与应用, 2021, 11(12): 2845-2852.
DOI: 10.12677/csa.2021.1112289

boundary conditions is established by using two-dimensional Saint Venant partial differential equations. The explicit finite difference method is used to discretize the model. Here, the central difference scheme is used in both time and space. The central difference scheme in time is also called leapfrog scheme. Then it is implemented and parallelized by MPI and OpenMP. For testing and experimental purposes, we used a simple cuboid tank to simulate an open channel. Through numerical simulation, the output parameters under different time steps, such as the height and speed of the water flow, are obtained, and then these parameters are processed to realize visualization. Finally, the parallel program is compared with the serial program, and the scale expansion test is carried out.

Keywords

Saint-Venant Equations, Shallow Water Equations, Finite Difference Method, Leapfrog Scheme, Parallel Computing, OpenMP

1. 介绍

水库大坝在防洪、供水、发电等方面发挥了巨大的作用，是水利工程体系中的重要组成部分。但是如果大坝因各种因素被损毁，那后果一般都是灾难性的。而超标准的洪水就是大坝溃决的自然因素之一。洪水是最频繁发生的自然灾害，影响了众多的人口和农田，而且还会造成人员伤亡和基础设施损坏。由于城市化，降雨时间延长和河流容量不足引起的径流率增加是洪水发生的一些主要原因。2020 年入汛以后，中国南方地区发生多轮强降雨过程，造成多地发生较重洪涝灾害。截至 7 月 9 日，洪涝灾害造成 27 省(区、市) 3020 万人次受灾，直接经济损失 617.9 亿元。2021 年，河南山西等地也遭受了特大暴雨的袭击。所以了解流道中水流的行为对于早期洪水灾难管理和挽救生命至关重要[1]。本文的研究方向就是通过对地表水流的数值模拟来研究洪水的行为，并对模拟过程进行并行处理和可视化。

数学模型是运用数理逻辑方法和数学语言建构的科学或工程模型，以解决各种各样的实际问题[2]。在水利科学中，就经常使用数学模型对不同现象的流体进行模拟，使用数值方法对流体模型进行控制。我们研究使用的是洪水波传播动力学方程，最早由圣维南提出[3]，所以也被称为 Saint-Venant (SV)方程。它由反映质量守恒定律的连续方程和反映动量守恒定律的运动方程组成，广泛用于预测地表流动参数，例如速度，深度或高度。SV 方程是针对一维的，对于二维的地表水流，SV 方程源自 Navier-Stokes 方程[4]，通常被称作浅水方程[5]。由于圣维南方程组在数学上属于拟线性双曲型偏微分方程组，很难用解析方法求得圣维南方程组的解析解。因此人们提出了不同的数值方法来进行地表水流的模拟。

在计算机出现之前，就有人通过数值模拟对圣维南方程进行求解，Reinaldo 和 Rene [6]较早地使用了基于显式 MacCormack 时间分裂格式，建立了求解二维 SV 方程组的数学模型，最后介绍了当时的两个工业应用，证明了模型的有效性。之后 Fiedler 和 Ramirez [7]也使用该方法模拟了具有空间可变性质的不连续二维水动力地表流动方程(二维圣维南方程的变体)。该方法被开发用于模拟空间可变渗透和微地形，也可用于模拟灌溉、潮滩和湿地循环以及洪水。之后随着计算机的发展，计算性能的提升[8]，越来越多的人将原本使用纸笔的模拟过程放到计算机上实现。比如 Kamboh [5]等人也是为了预测和模拟洪水行为，使用二维圣维南偏微分方程建立了具有初始条件和边界条件的数学模型。接下来，使用常见的显式有限

差分法将相应模型离散化并在 MATLAB 上实现。最终生成的图形结构可以直观地看到随时间变化的各参数变化情况。Asier [9]等人同样是使用二维圣维南方程来模拟降水或径流事件，使用的方法却是有限体积法，按照以下三种编程方法进行了开发和比较：顺序、多线程和众核架构。多线程代码是使用 OpenMP 库编写的，而众核架构是使用 CUDA 编写的。

综上所述，虽然很多不同的系统已经针对 SV 方程进行建模，并且还有了并行化、性能优化的讨论，但是还很少有对大规模圣维南系统的研究。因此本研究旨在建立一个二维的简单的有限差分模型，并使用 MPI + OpenMP 进行大规模处理。

2. 系统介绍

2.1. 控制方程

二维圣维南方程有很多不同的形式，因为我们并不是研究不同条件下圣维南方程不同的表示，所以只是使用了一个较为常见和简单的针对明渠的表示形式，公式如下所示，

$$\frac{\partial z}{\partial t}+\frac{\partial (zu)}{\partial x}+\frac{\partial (zv)}{\partial y}=0, \tag{1}$$

$$\frac{\partial (zu)}{\partial t}+\frac{\partial \left(zu^2+gz^2/2\right)}{\partial x}+\frac{\partial (zuv)}{\partial y}=gz\left(S_{0x}-S_{fx}\right), \tag{2}$$

$$\frac{\partial (zv)}{\partial t}+\frac{\partial (zuv)}{\partial x}+\frac{\partial \left(zv^2+gz^2/2\right)}{\partial y}=gz\left(S_{0x}-S_{fx}\right). \tag{3}$$

方程 1 是由质量守恒得来，方程 2 和 3 分别来自 x，y 两个方向上的动量守恒。其中，z 是指的明渠水流的海拔(深度或高度)，u、v 分别是在 x，y 两个方向上的水的流速，t 是时间，g 是重力加速度，S_0 和 S_f 是指水面梯度和摩擦阻力，下标中的 x，y 也同样表示不同方向。为了在计算机中更方便的使用这个方程组，我们需要使用微分的乘积法则对公式(1) (2) (3)进行变形，之后使用显示有限差分格式进行离散化，其中时间和空间导数分别由以下表达式离散化：

$$\frac{\partial u}{\partial t}\approx\frac{u_{i,j}^{k+1}-u_{i,j}^{k-1}}{2\Delta t},\frac{\partial u}{\partial x}\approx\frac{u_{i+1,j}^{k}-u_{i-1,j}^{k}}{2\Delta x},\frac{\partial u}{\partial y}\approx\frac{u_{i,j+1}^{k}-u_{i,j-1}^{k}}{2\Delta y}$$

时间和空导数都是使用中心差分格式，其中时间上的中心差分也被称为跳蛙格式，它的优点是可以加强计算的稳定性。这样离散化后的等式就变成下列这种形式，

$$\frac{z_{i,j}^{k+1}-z_{i,j}^{k-1}}{2\Delta t}+u_{i,j}^{k}\frac{z_{i+1,j}^{k}-z_{i-1,j}^{k}}{2\Delta x}+z_{i,j}^{k}\frac{u_{i+1,j}^{k}-u_{i-1,j}^{k}}{2\Delta x}+v_{i,j}^{k}\frac{z_{i,j+1}^{k}-z_{i,j-1}^{k}}{2\Delta y}+z_{i,j}^{k}\frac{v_{i,j+1}^{k}-v_{i,j-1}^{k}}{2\Delta y}=0, \tag{4}$$

$$\begin{aligned}&u_{i,j}^{k}\frac{z_{i,j}^{k+1}-z_{i,j}^{k-1}}{2\Delta t}+z_{i,j}^{k}\frac{u_{i,j}^{k+1}-u_{i,j}^{k-1}}{2\Delta t}+\left(u_{i,j}^{k}\right)^2\frac{z_{i+1,j}^{k}-z_{i-1,j}^{k}}{2\Delta x}+2z_{i,j}^{k}u_{i,j}^{k}\frac{u_{i+1,j}^{k}-u_{i-1,j}^{k}}{2\Delta x}+gz_{i,j}^{k}\frac{z_{i+1,j}^{k}-z_{i-1,j}^{k}}{2\Delta x}\\&+u_{i,j}^{k}v_{i,j}^{k}\frac{z_{i,j+1}^{k}-z_{i,j-1}^{k}}{2\Delta y}+z_{i,j}^{k}v_{i,j}^{k}\frac{u_{i,j+1}^{k}-u_{i,j-1}^{k}}{2\Delta y}+z_{i,j}^{k}u_{i,j}^{k}\frac{v_{i,j+1}^{k}-v_{i,j-1}^{k}}{2\Delta y}=gz_{i,j}^{k}\left(S_{0x}-S_{fx}\right),\end{aligned} \tag{5}$$

$$\begin{aligned}&v_{i,j}^{k}\frac{z_{i,j}^{k+1}-z_{i,j}^{k-1}}{2\Delta t}+z_{i,j}^{k}\frac{v_{i,j}^{k+1}-v_{i,j}^{k-1}}{2\Delta t}+\left(v_{i,j}^{k}\right)^2\frac{z_{i,j+1}^{k}-z_{i,j-1}^{k}}{2\Delta y}+2z_{i,j}^{k}v_{i,j}^{k}\frac{v_{i,j+1}^{k}-v_{i,j-1}^{k}}{2\Delta y}+gz_{i,j}^{k}\frac{z_{i,j+1}^{k}-z_{i,j-1}^{k}}{2\Delta y}\\&+u_{i,j}^{k}v_{i,j}^{k}\frac{z_{i+1,j}^{k}-z_{i-1,j}^{k}}{2\Delta x}+z_{i,j}^{k}v_{i,j}^{k}\frac{u_{i+1,j}^{k}-u_{i-1,j}^{k}}{2\Delta x}+z_{i,j}^{k}u_{i,j}^{k}\frac{v_{i+1,j}^{k}-v_{i-1,j}^{k}}{2\Delta x}=gz_{i,j}^{k}\left(S_{0y}-S_{fy}\right).\end{aligned} \tag{6}$$

这样在简单的移项、去分母后，就可以迭代求解 z、u、v 三个变量。如图 1 所示，每次计算一个迭代值的时候都要使用到上两次迭代的结果，这也是跳蛙格式的特性。也正因如此，加强了计算过程的稳定性。

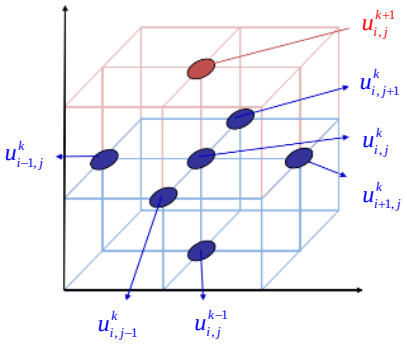

Figure 1. Schematic diagram of iterative process
图 1. 迭代过程示意图

2.2. 初始条件

考虑到先将并行逻辑完整讲述，再进行规模扩大，我们采用如图 2 一样的长方体水槽来模拟单一明渠内流体的流动行为。

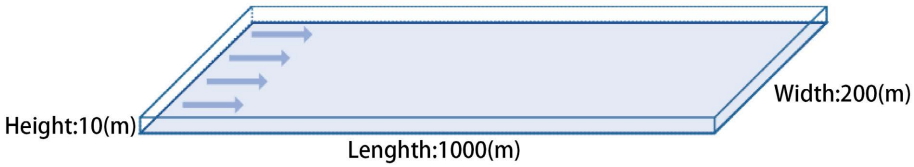

Figure 2. Schematic diagram of single open channel
图 2. 单个明渠示意图

最初，我们让整个区域处于静止状态，即 $z = 10$ m，$u = 0$ m/s，$v = 0$ m/s，然后使用公式

$$z = \mathrm{e}^{-\frac{(a-a_0)^2+(b-b_0)^2}{k^2}}$$

在明渠的中心点产生高度为 1 m 的洪水波。这个公式中，a_0 和 b_0 分别是泛洪波最高处在 x 方向 y 方向上的定位，k 是最初始的水波高度。至于 a 和 b 则是每个网格节点在坐标轴上的坐标。在我们的例子中，我们假设在区域中心产生洪水波，并且还考虑了 ghost cell，因此测试值取为 $a = 100$，$b = 500$，$k = 10$。

除上述变量外，还有一些在这里展示，Δx 和 Δy 均为 1 m，$\Delta t = 0.01$ s，且 $S_{0x} = 0$，$S_{fx} = 0.1$，$S_{0y} = 0$，$S_{fy} = 0.1$。还有我们不考虑复杂的边界条件，选择使用自反射边界条件。

2.3. 并行策略

如同 2.2 提到的同样的原因，我们先选择使用四个节点进行小规模并行化。我们使用 MPI 对每个节点进行操作。这样，我们把 200 * 1000 大小的网格划分为四个大小同样为 100 * 500 的小网格，每个节点拥有一个自定义结构体存储每一次迭代每一个网格节点内 z, u, v 的值。我们使用 OpenMP 中#pragma omp parallel for 对单次迭代中的结构为双层循环的计算部分进行并行化，对计算结束后的两个结构为单层循环的边界处理也进行同样处理。因为图 1 所展示的计算过程，使得在计算小网格边界值时需要使用相邻网

格的边界值，所以我们使用 MPI_Sendrecv()函数进行节点之间的数据交换。在数据交换结束后，一次迭代才算作完成。通过图 3、图 4 可以更直观地理解这一并行处理。

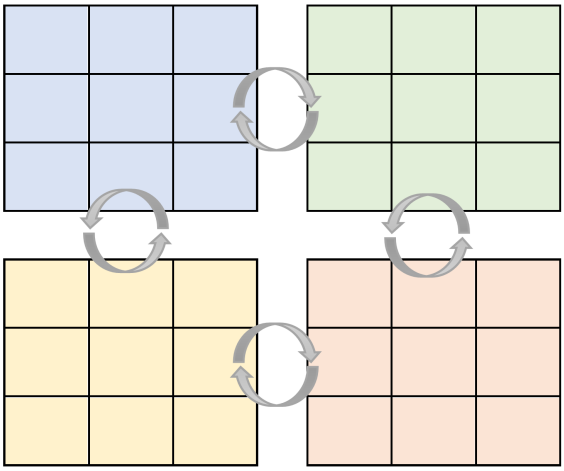

Figure 3. Schematic diagram of grid data exchange
图 3. 网格数据交换示意图

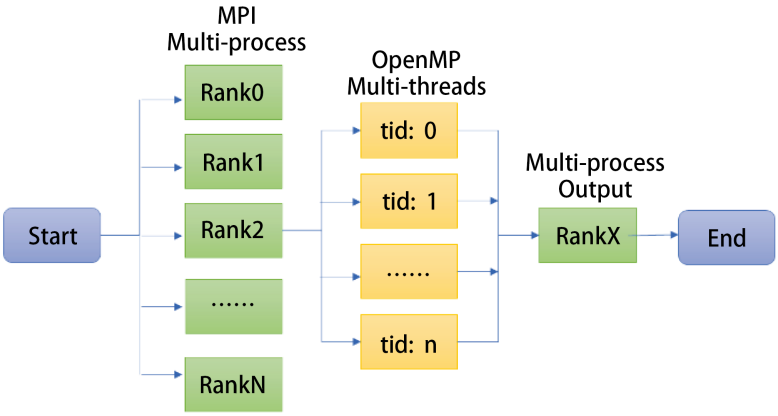

Figure 4. Parallel framework
图 4. 并行框架

2.4. 结果

我们将每次迭代每个网格节点的 z，u，v 值全部输出到逗号分隔值文件中，再对文件进行处理并可视化，就能很直观的看到实验结果。图 5 所展示的就是在不同迭代次数下模拟的二维 SV 方程的海拔，通过它能够清楚看出随时间变化水面高度变化的情况。图中第一个曲面所展示的是开始迭代之前的状态，经过迭代计算不断地变换成后续的状态。同样的我们也可以去分析 u，v 的图像，但因其看起来不够直观，在此不予展示。

3. 大规模与性能分析

我们的并行框架在设计之初就是为了大规模模拟而设计的，只需要简单更改初始条件和节点规模，规模就可以不断扩大。通过宏定义，节点内的计算规模、节点逻辑上的排列方式、节点间通信地址的寻址方式都是可以随时更改的，所以我们可以很轻易的将我们模型的规模不断扩大。

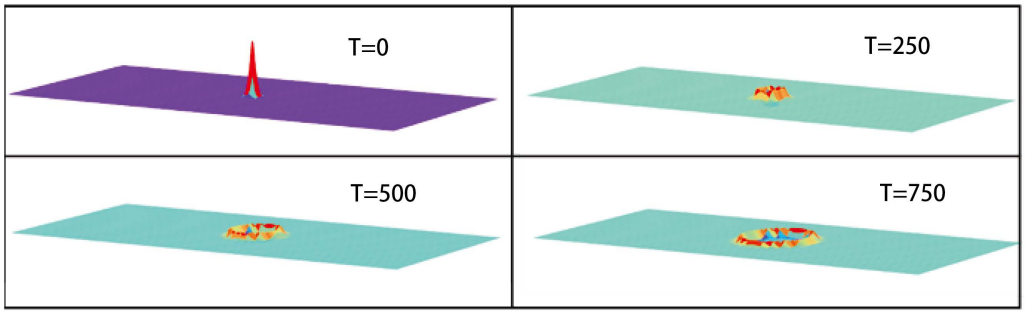

Figure 5. Simulation of water surface height of open channel
图 5. 明渠水面高度模拟

接下来就是对大规模扩展进行测试，我们所有测试都是运行在国家超级计算昆山中心的共享超级计算平台。本计算集群每个节点配置 1 颗 32 核心 2.5 GHz 主频的 x86 处理器和 8 根 16 GB DDR4 2666 ECC REG 内存，采用全线速、无阻塞的 200 Gb HDR Infiniband 专用计算网络。而 MPI 是用 hpcx-2.4.1 编译的，OpenMP 启用的线程数是 32 个。初始化过程和最终输出过程虽然做了多进程并行优化，但不计入执行时间。我们在大规模方面进行了强弱扩展测试。首先是弱扩展测试，在保持每个进程分配的计算量基本一致的情况下，通过不断增加进程数来测试程序的并行性能，图 6 是弱扩展测试的折线图，图 7 是对应的并行效率；之后是强扩展测试，在保证问题计算规模一定的情况下，测试程序随着核数增多单个节点计算量降低而导致程序性能的变化情况，如图 8 所示。此处性能的计算公式为：

$$S_p = \frac{T_1}{T_p}, E = \frac{S_p}{p}$$

式中：p 为 CPU 数，即节点数，T_1 为串行执行算法的执行时间，T_p 为 p 个处理器时并行算法的执行时间。

我们强弱扩展测试均以单核时间为基准，每种情况测试 10 次取均值，迭代次数为 1000 次。弱扩展测试的并行效率在千核时还保持在 80%，框架始终运行在较高的效率之上，展性较好。强展性测试也有较为理想的并行效率，在千核内同样保持 80%效率，说明划分平衡度较好。

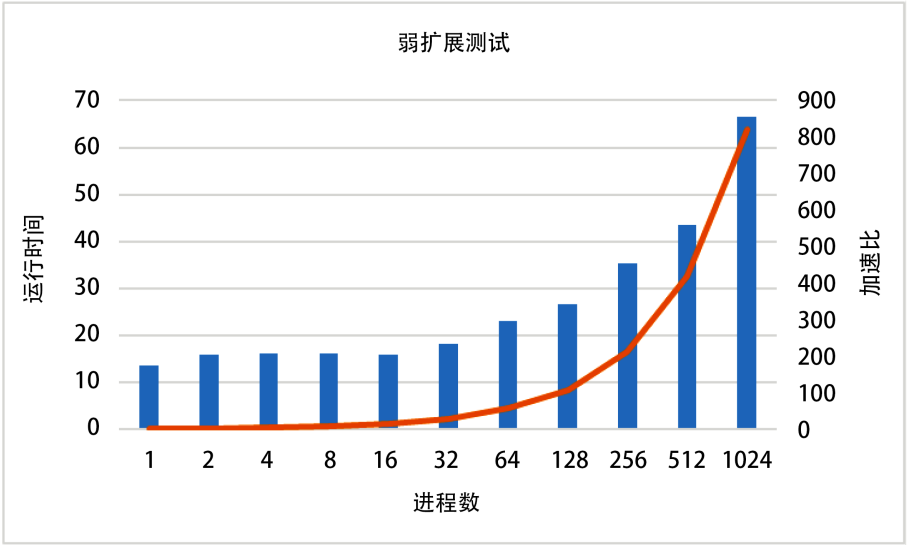

Figure 6. Weak expansion time and acceleration ratio
图 6. 弱扩展测试时间与加速比

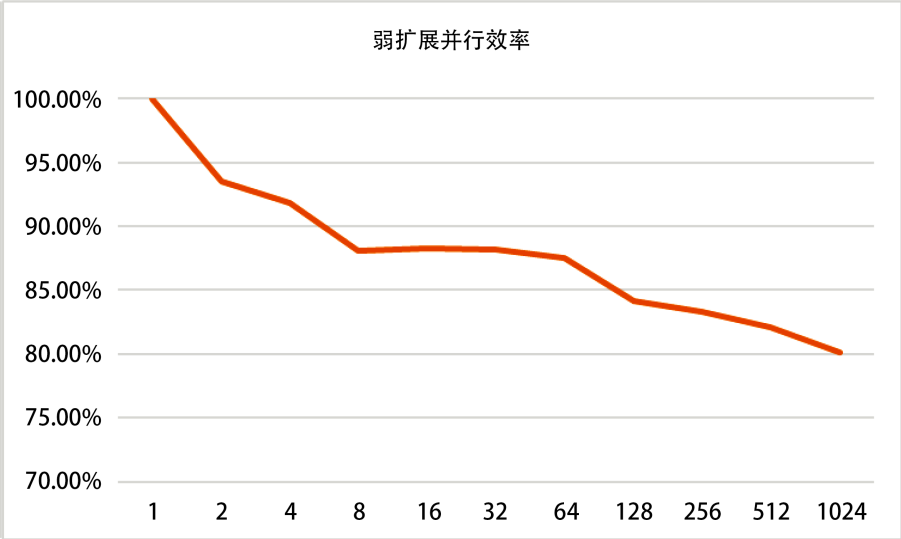

Figure 7. Parallel efficiency
图 7. 弱扩展测试并行效率

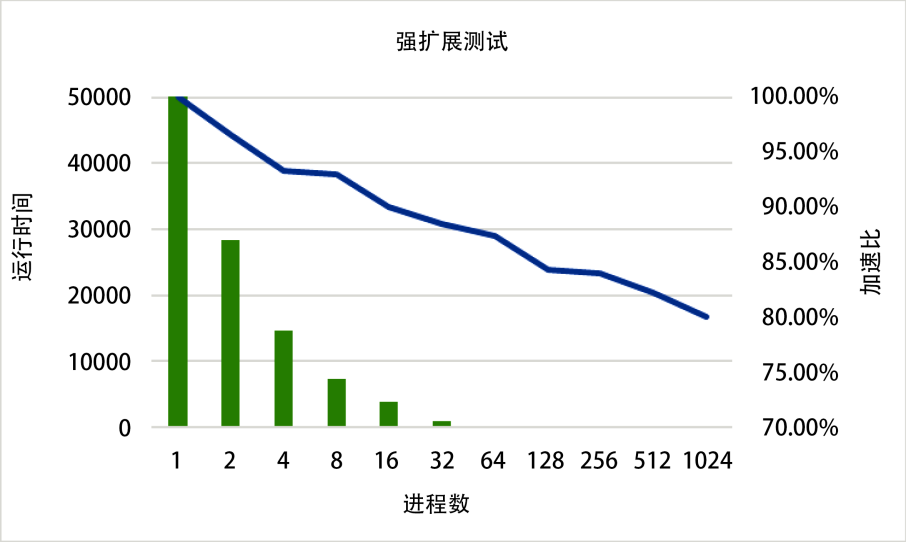

Figure 8. Strong expansion test
图 8. 强扩展测试

4. 总结

本文使用数值方法来模拟明渠中的二维圣维南系统。使用 Leapfrog 方案的有限差分法求解方程。考虑到大规模并行化要求，考虑了一个简单的长方体明渠，在中心产生洪水波。通过对 z，u，v 值的迭代计算输出，得到可视化图形，增强人们对洪水波的认知。然后使用 MPI + OpenMP 对程序进行并行化，分析并行效率。最后得出结论，所研究的并行框架具有良好的稳定性和可扩展性。

参考文献

[1] Lv, Z., Li, J., Dong, C., *et al*. (2021) Deep Learning in the COVID-19 Epidemic: A Deep Model for Urban Traffic Revitalization Index. *Data & Knowledge Engineering*, **135**, 101912.

[2] Lv, Z., Li, J., Li, H., *et al*. (2021) Blind Travel Prediction Based on Obstacle Avoidance in Indoor Scene. *Wireless Communications and Mobile Computing*, **2021**, Article ID 5536386. https://doi.org/10.1155/2021/5536386

[3] Saint-Venant, B.D. (1871) Theory of Unsteady Water Flow, with Application to River Floods and to Propagation of Tides in River Channels. *French Academy of Science*, **73**, 148-154, 37-240.

[4] Dawson, C. and Mirabito, C.M. (2008) The Shallow Water Equations.

[5] Kamboh, S.A., Kamboh, S.A., Sarbini, I.N., Labadin, J. and Monday, E.O. (2015) Simulation of 2d Saint-Venant Equations in Open Channel by Using Matlab. *Journal of IT in Asia*, **5**, 15-22. https://doi.org/10.33736/jita.47.2015

[6] Garcia, R. and Kahawita, R.A. (2010) Numerical Solution of the st. venant Equations with the Maccormack Finite-Difference Scheme. *International Journal for Numerical Methods in Fluids*, **6**, 259-274. https://doi.org/10.1002/fld.1650060502

[7] Fiedler, F.R. and Ramirez, J.A. (2000) A Numerical Method for Simulating Discontinuous Shallow Flow over an Infiltrating Surface. *International Journal for Numerical Methods in Fluids*, **32**, 219-239. https://doi.org/10.1002/(SICI)1097-0363(20000130)32:2<219::AID-FLD936>3.0.CO;2-J

[8] Lv, Z., Li, J., Xu, Z., *et al.* (2021) Parallel Computing of Spatio-Temporal Model Based on Deep Reinforcement Learning. *International Conference on Wireless Algorithms, Systems, and Applications*, Springer, Cham, 391-403. https://doi.org/10.1007/978-3-030-85928-2_31

[9] Lacasta, A., Morales-Hernandez, M., Murillo, J. and Garcia-Navarro, P. (2015) Gpu Implementation of the 2d Shallow Water Equations for the Simulation of Rainfall/Runoff Events. *Environmental Earth Sciences*, **74**, 7295-7305. https://doi.org/10.1007/s12665-015-4215-z