Computers & Geosciences 78 (2015) 110–122

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

Regional-scale calculation of the *LS* factor using parallel processing

Kai Liu [a,b,c], Guoan Tang [a,b,c,*], Ling Jiang [d], A-Xing Zhu [c,e], Jianyi Yang [a,b,c], Xiaodong Song [f]

[a] Key Laboratory of Virtual Geographic Environment, Nanjing Normal University, Ministry of Education, Nanjing 210023, China
[b] State Key Laboratory Cultivation Base of Geographical Environment Evolution, Jiangsu Province, Nanjing 210023, China
[c] Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, Nanjing 210023, China
[d] Anhui Center for Collaborative Innovation in Geographical Information Integration and Application, Chuzhou University, Chuzhou 239000, China
[e] Department of Geography, University of Wisconsin-Madison, Madison, USA
[f] State Key Laboratory of Soil and Sustainable Agriculture, Institute of Soil Science, Chinese Academy of Sciences, Nanjing, Jiangsu 210008, China

ARTICLE INFO

Article history:
Received 22 August 2014
Received in revised form
20 December 2014
Accepted 2 February 2015
Available online 11 February 2015

Keywords:
LS factor
Parallel computing
Soil erosion model
Regional scale
Digital terrain model

ABSTRACT

With the increase of data resolution and the increasing application of USLE over large areas, the existing serial implementation of algorithms for computing the *LS* factor is becoming a bottleneck. In this paper, a parallel processing model based on message passing interface (MPI) is presented for the calculation of the *LS* factor, so that massive datasets at a regional scale can be processed efficiently. The parallel model contains algorithms for calculating flow direction, flow accumulation, drainage network, slope, slope length and the *LS* factor. According to the existence of data dependence, the algorithms are divided into local algorithms and global algorithms. Parallel strategy are designed according to the algorithm characters including the decomposition method for maintaining the integrity of the results, optimized workflow for reducing the time taken for exporting the unnecessary intermediate data and a buffer–communication–computation strategy for improving the communication efficiency. Experiments on a multi-node system show that the proposed parallel model allows efficient calculation of the *LS* factor at a regional scale with a massive dataset.

1. Introduction

Among the factors in the Universal Soil Loss Equation (USLE) (Wischmeier and Smith, 1978) and the revised USLE (RUSLE) (Renard et al., 1997), the extraction of the *LS* factor, which reflects the influence of terrain on soil erosion, is a key issue in the applications of these models (Kinnell, 2010). The *LS* factor contains two parts: the slope length factor (*L*) and the slope factor (*S*). It is widely believed that slope length is the more problematic part. The development of GIS allows for automatic extraction of slope length from high resolution DEMs, thus an inefficient manual process is avoided. In general, there are three kinds of extraction methods: unit stream power methods (Moore and Burch, 1986; Mitasova, 1996), contributing area methods (Moore and Wilson, 1992; Desmet and Govers, 1996) and grid-based methods (Hickey, 1994, 2000). In recent years, researches have primarily adopted grid-based methods, which overcome the limitation of the former two methods in predicting soil deposition on topographically driven zones (Winchell et al., 2008).

In the grid-based method proposed by Hickey et al. (1994), slope length is calculated from the high points along the direction with maximum downhill slope angle at the same time that converging flows and deposition areas are taken into consideration. Based on the work of Van Remortel et al. (2001, 2004), Zhang et al. (2013) proposed a definition of distributed watershed erosion slope length (DWESL) with the following two improvements: (1) for the runoff node, the slope length equals to the total lengths of all the surrounding cells flowing into it, instead of just considering the longest value; and (2) the slope length calculation must stop at a channel. The experiments showed that the results got by DWESL method were more similar to the manual method than those by existing algorithms.

In many studies, USLE/RUSLE has been employed for the quantitative assessment of soil erosion at large watersheds or even on a regional scale (Yang et al., 2003; Fu et al., 2005). At the same time, the development of acquisition method for spatial data means that the geospatial dataset size is increasing at an exponential rate (Cheng et al., 2013) and high resolution DEM data for large regions are becoming more available. As a result, the computational time increases significantly and to the level becoming a bottleneck for applying these models over large areas

* Corresponding author at: Key Laboratory of Virtual Geographic Environment, Nanjing Normal University, Ministry of Education, Nanjing 210023, China. Fax: +86 25 85891347.
E-mail addresses: liukaigis@gmail.com (K. Liu), tangguoan@njnu.edu.cn (G. Tang).

http://dx.doi.org/10.1016/j.cageo.2015.02.001
0098-3004/

(Jiang et al., 2013). It is limited by computer storage and calculation ability that traditional serial processing cannot meet user demand due to the long response time. Currently, two methods are adopted for *LS* factor calculation on regional scale. One method employs a set of prediction rules to generate *LS* factor based on a number of attributers which control the landform types such as materials, climate and regional geomorphology (Lu et al., 2003). Another extracts the topographic index based on low resolution DEMs, and then gets available values via a scale transformation (Cheng et al., 2009). However, both methods could only be regarded as an expedient measure, the fast and accurate calculation of *LS* factor on regional scale is yet to be developed.

Developments in computer technology have improved computation ability by using parallel processing. Recently, parallel technology has been widely used in digital terrain analysis, such as parallel drainage extraction (Gong and Xie, 2009; Qin and Zhan, 2012), parallel visibility analysis (Kidner et al., 1997; Wang et al., 2015), and parallel hydrological analysis (Tesfa et al., 2011; Wu et al., 2013; Liu et al., 2013). In order to make the parallel programming become easy, some raster-based programming libraries are proposed such as Parallel Raster Processing Programming Library (pRPL) (Guan and Clarke, 2010) and Parallel Raster-based Geocomputation Operators (PaRGO) (Qin et al., 2014). Among the existing literatures, Message Passing Interface (MPI) is widely used to parallelize algorithms for its adaptable to various parallel computing environments and the rich programming interfaces (Cheng et al., 2013). Compared with the serial algorithms, the MPI parallel algorithms can achieve a huge improvement in processing time (Jiang et al., 2013). There is, however, little published research on the *LS* factor calculation using parallel computing.

The aim of this paper is to propose a parallel approach that can be applied on a regional scale calculation of the *LS* factor. The structure of this paper is as follows. Section 2 details each algorithm for the *LS* factor calculation. Section 3 introduces the parallel approach. According to the algorithm characters, two parallel strategies have been designed for both local algorithms and global algorithms. Section 4 discusses the effectiveness of the parallel algorithms, and finally, the concluding remarks are given in Section 5.

2. Serial *LS* factor calculation algorithm

2.1. Overview of the LS factor algorithms

The *LS* factor can be obtained according to the following expressions (McCool et al., 1989; Liu et al., 2000):

$$LS = L*S \tag{1}$$

$$L = \left(\frac{l}{22.13}\right)^m \tag{2}$$

$$\lambda_{i,j} = \sum_{x=0,y=0}^{x=i,y=j} \sum_{k=1}^{m} \lambda_{x,y} \tag{3}$$

$$\beta = (\sin\theta/0.089)/[3\sin\theta^{0.8} + 0.56] \tag{4}$$

$$S = \begin{cases} 10.8\sin\theta + 0.03 & \theta < 5^\circ \\ 16.8\sin\theta - 0.05 & 5^\circ \le \theta \le 14^\circ \\ 21.91\sin\theta - 0.96 & 14^\circ \le \theta \end{cases} \tag{5}$$

where l is the slope length, m is a variable length-slope exponent, β is the ratio between rill erosion and interrill erosion, θ is the angle of the slope, S is the slope factor and L is the slope length factor.

In the calculation of *LS* factor, the slope algorithm is relatively simple, while the slope length algorithm is more complicated. In this paper, the DWESL method proposed by Zhang et al. (2013) is used and its formula is as follows:

$$\lambda_{i,j} = \sum_{x=0,y=0}^{x=i,y=j} \sum_{k=1}^{m} \lambda_{x,y} \tag{6}$$

where k is the code of the surrounding cells at coordinates (x, y), and $\lambda_{x,y}$ is the slope length of each cell.

According to Eq. (6), for the cells where convergent flow occurs, the slope length of the center cell should equal to the sum value of the slope length of the surrounding cells that flow into it, which shows the particularity between DWESL and the other grid-based approaches. As shown in Fig. 1, point A is the convergent point of flow AC and BC. The slope length of A takes the sum value of projection length $A'B'$ and $A'C'$ instead of using the longest length. In addition, two cutoff factors are considered in DWESL. One is slope cutoff factor, such as Point D, where slope becomes gentle and deposition happens. In this case, the calculation of slope length should be restarted. The other one is the channel network cutoff factor. At Point F, where channel network occurs, the slope length should be set to a constant, generally zero.

2.2. Algorithm flow

The algorithm flow is based on the definition above and contains the six steps shown in Fig. 2. The input DEM data need to be preprocessed before the calculation in order to fill surface depressions and remove flat areas. In this paper, the preprocessed processing has been implemented using TauDEM software (Wallis et al., 2009).

2.3. Flow accumulation calculation

Among the calculation workflow above, the algorithms for flow accumulation and slope length calculations are more complex due to their accumulation process. It is obvious that the flow accumulation calculation for each cell depends on its upstream cells. In other word, to calculate the flow accumulation, the number of all cells that drain into it directly or indirectly should be figured up (Wallis et al., 2009). In the process, all cells are actually composed of two kinds of cells: (1) source cell, located in the peak or ridge or around the edge of DEM dataset with no cell drain into; and (2) normal cell, which does not belong to source cells (Fig. 3a). The serial algorithm for flow accumulation is as follows:

Step 1: Each cell calculates the number of neighbor cells which flow into it. If the number equals to n, then cell is assigned to $-n$; if the number equals to 0, then the cell value is assigned to zero and it should be pushed onto the stack.

Step 2: Each cell in the stack should be popped off in turn, if its next cell n in flow direction is less than -1, the value of cell n pluses one and waits for the next calculation. If its next cell n in flow direction equals to -1, which means the computational condition is achieved, cell n is assigned to the sum value of neighbor cells. Iterating the calculation for each cell from cell n until the next cell does not equal to -1.

Step 3: As soon as the stack is getting empty, which means all the cells have been assigned to its flow accumulation value, then the computation task can be completed.