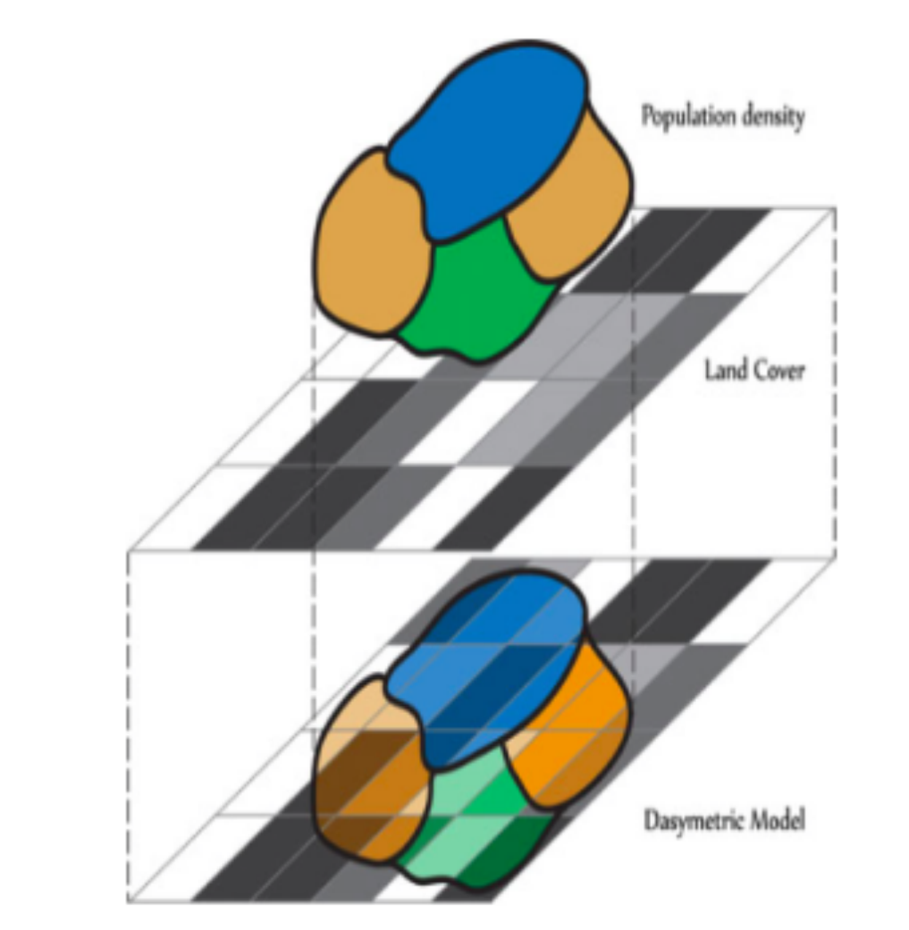
Parallel Dasymetric Mapping in GIS Modeling

Zhen Zhang(City University of Hong Kong)
Mentors: Lonnie Crosby, Cheng Liu, Nicolas Nagle, Kwai Wong(UT)

Abstract

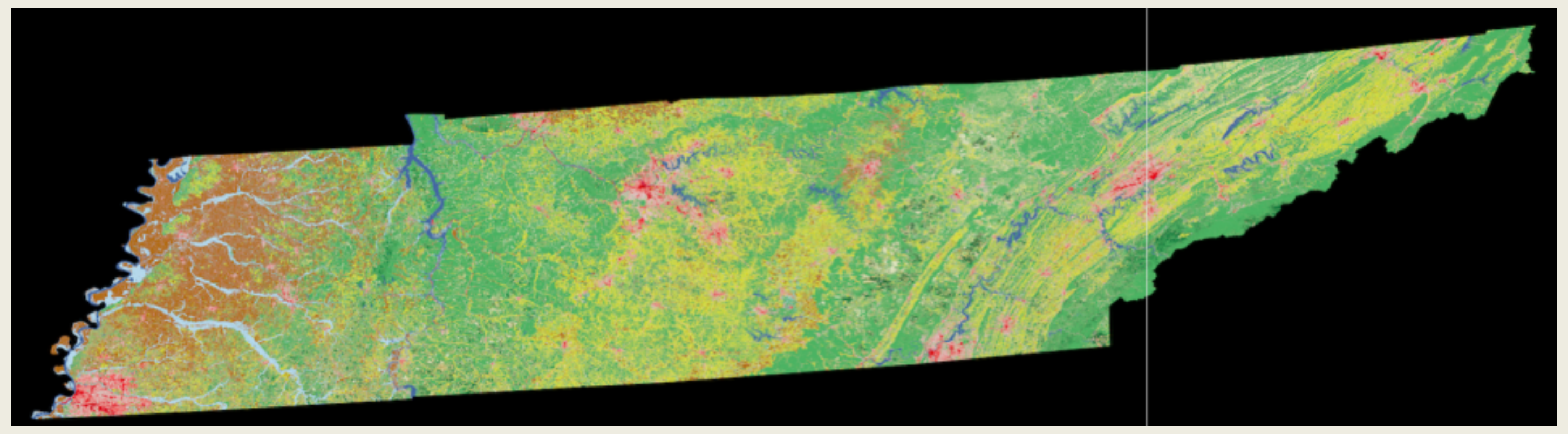
Dasymetric mapping is a commonly-applied algorithm in GIS that can determine the population distribution in a high resolution level. It maps the population data with a coarse resolution with a high-resolution ancillary-level data (most of the time, it's land type data). However, for a large-scale problem, it may take a long time to do this process. So we will propose a parallel method for this problem. And we will use this method to perform dasymetric mapping on Tennessee. If the result is successful, we can expand the source area to the whole U.S.



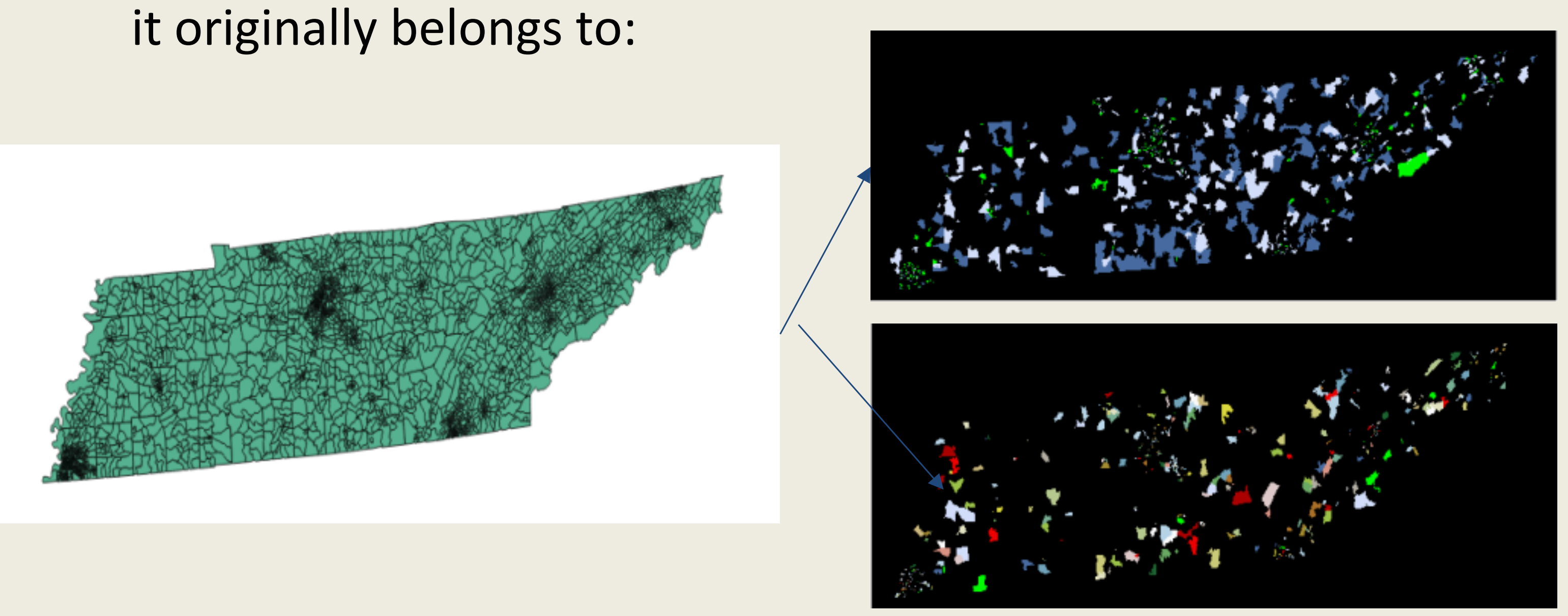
Data Preparation

We start from 3 datasets. First, NLCD data, which contains land-type information on 30m*30m level for the whole U.S. Second, boundary data of block groups in a state, which is the shape of boundary of different block groups.(Block group is a small census district unit.) Third, the ACS Summary file which tells the population in each block group. They are all available on the website. These are the 3 levels of input data. They need to be changed to be consistent with each other.

First we can extract NLCD data set of a given state from the whole U.S. using QGIS(a software for geographical information system). It looks like this:



Then, we need to rasterize the boundary file file of block groups so that we can map it with NLCD data together. Rasterization is a process that changes polygons into a big matrix, where each cell of the matrix tells which block group it originally belongs to:



The original shape file is rasterized into 2 matrices, because Tennessee has 4125 block groups, which can not be represented using 1 byte. So the first matrix contains the first byte of block group ID, while the second matrix contains the second byte of block group ID.

Last, we have to extract the total population of each block group from ACS Summary file, and sort them by block group ID. This step can be done using R.

Brief Estimate

Now we have the land type data for Tennessee, however, it cannot be used directly to determine population distribution. However, it can be used to give a brief estimate of population density in each 30m*30m cell. We can use a linear regression model to model the relation between land type and density:

$$P_S = \alpha + \sum_{c=1}^{C} \beta_c A_{SC} + \epsilon_S$$

where P_S is the population of zone S; α is the intercept term; β_c is the coefficient for land cover c; A_{sc} is the area of land cover c within zone s; C is the number of populated land cover types occurring within study region; and ϵ_S is a random error term.
It can also be written as: $E(P_S|A_S) = \alpha + \beta A_S$
where β and A_S are vectors. Negative population totals can also be avoided by using Poisson regression:

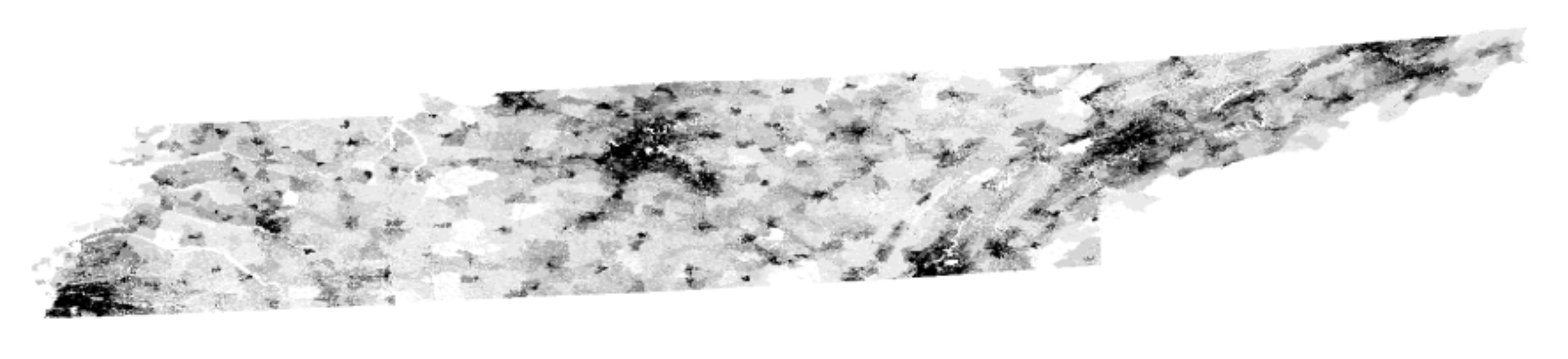
$$E(P_S|A_S) = \exp(\alpha + \beta A_S)$$

Here S(source zone) is a block group. The values in β are the estimated population density for cells with different land types.

Calculation

Create an empty matrix W. Then assign a weight to each cell In proportional to the estimated density. We can count the total weights in a block group. Then the more precise population estimation for a cell will be the total population of the block group that the cell belongs to times the weight of this cell divided by the total weights this block group. After this calculation, we can get a matrix in the same size as the state-level NLCD dataset. Each entry of this matrix tells the population density of itself.

One thing to notice is that due to our target region is very small, certain approximation or aggregation is needed when making real maps.



(This picture is generated according to the method given above. Darker means higher population density)

Parallelization

We would like to use distributed memory to implement the parallel method, so a state will be cut into several boxes. Two parts will be different from serial method. The first part is in determining the brief estimate of population density. Each box will calculate the β for itself. The values in β can be different for different boxes. Also, in the regression process, we need the total area of land cover c within zone s for every block group. However, for those block groups on the intersection between boxes, we cannot know this value for them. Thus we will find these block groups and make sure they will not participate in the regression. The second part is that we need to transfer the weight of cells of block groups on the intersection between boxes to its neighborhood to get the total weight of that block group.

Due to more than 90% or more block groups will be 'internal' area(not on the intersection), and their population density can be calculated using the same method as the serial one inside each box, the data transferring on the boundary will not take a long time in general.

Future Work

For sure, the method above is not the only method. We can try polygon method instead of grid method. Also we plan to implement Prof. Nagle's PMEDM Method for dasymetric mapping and adapt it to become parallel. We will also extend our studying area into the whole U.S. After that, we will make analysis on the improvement of efficiency using parallel computing compared to doing serially.

Acknowledgements

Support from City University of Hong Kong, Oak Ridge National Laboratory, Joint Institute for Computational Sciences and University of Tennessee are gratefully acknowledged.