17th International Conference
ENGINEERING MECHANICS 2011
Svratka, Czech Republic, 9 – 12 May 2011

FAST EVALUATION OF LINEAL PATH FUNCTION USING GRAPHICS PROCESSING UNIT

J. Havelka[*], J. Sýkora[*], A. Kučerová[*]

Abstract: *Homogenization methods are becoming the most popular approach to modelling of heterogeneous materials. The main principle is to represent the heterogeneous microstructure with an equivalent homogeneous material. When dealing complex random microstructures, the unit cell representing exactly periodic morphology needs to be replaced by a statistically equivalent periodic unit cell (SEPUC) preserving the important material properties in the statistical manner. One of the statistical descriptors suitable for SEPUC definition is the lineal path function. It is a low-order descriptor capable of capturing certain information about the phase connectedness. Its main disadvantage is the computational cost. In this contribution, we present the reformulation of the sequential C code for evaluation of the lineal path function into the parallel C code with Compute Unified Device Architecture (CUDA) extensions enabling the usage of computational potential of the NVIDIA graphics processing unit (GPU).*

Keywords: ***Lineal path function, homogenization, statistically equivalent periodic unit cell, graphics processing unit.***

1. Introduction

Modelling of random heterogeneous materials is a multi-disciplinary problem with a wide range of relevant engineering applications. The unifying theoretical framework is provided by homogenization theories, which aim at the replacement of the heterogeneous microstructure with an equivalent homogeneous material, e.g. (Torquato, 2002). Currently, two main approaches are available: (i) computational homogenization and (ii) effective media theories. While the first class of methods studies the distribution of local fields within a typical heterogeneity pattern using a numerical method, the second group estimates the response analytically on the basis of limited geometrical information (e.g. the volume fractions of constituents) of the analysed medium.

It is generally accepted that detailed discretization techniques, and the Finite Element Method (FEM) in particular, remain the most powerful and flexible tools available. Despite of the tedious computation time, it provides us details of local stress and strain fields. Moreover, it is convenient to characterize the material heterogeneity by introducing the concept of a Periodic Unit Cell (PUC) (Vorel, 2009) or Statistically Equivalent Periodic Unit Cell (SEPUC), see (Zeman & Šejnoha, 2007; Vorel et al., 2011) for more details. On the other hand, if only the overall (macroscopic) response is demanded variable, it is sufficient to introduce structural imperfections in a cumulative sense using one of the averaging schemes, e.g. the Mori-Tanaka method (Vorel & Šejnoha, 2009).

If the effective material parameters of complex microstructure (see Fig. 1) are demanded, the homogenization technique based on the SEPUC can be utilized. Furthermore, this approach allows us to reduce the computation cost by generating smaller unit cell describing the real structure. The generation of the SEPUC is based on optimization of an appropriate statistical descriptor. The most commonly used group of descriptors embodies a set of general n-point probability functions, applicable to an arbitrary two-phase composite (Torquato, 2002). A different statistical function deserves attention when phase connectivity information is to be captured in more detail, as e.g. for medium at Figure 1. Therefore we focus here on usage of the lineal path function. The principal drawback concerns its evaluation, which is non-negligible time-consuming, especially when evaluated

[*] Jan Havelka, Ing. Jan Sýkora, Ph.D. and Ing. Anna Kučerová, Ph.D.: Faculty of Civil Engineering, Czech Technical University in Prague, Thákurova 7/2077; 166 29, Prague; CZ, e-mails: jan.havelka.1@fsv.cvut.cz, jan.sykora.1@fsv.cvut.cz, anicka@cml.fsv.cvut.cz

many times within the optimization process. Hence, we present an accelerated implementation of the lineal path function on the GPU.

Fig. 1: Three cuts through trabecular bone microstructure obtained by micro Computed Tomography (Jiroušek et al., 2008), courtesy of O. Jiroušek, Institute of theoretical and applied mechanics.

The following section contains the definition of the lineal path function. The Section 3 discusses its algorithmic formulation and Section 4 presents the resulting speed-up obtained at GPU in comparison with the sequential CPU formulation together with concluding remarks.

2. Lineal path function

The lineal path function (Lu & Torquato, 1992) is one of the low-order microstructural descriptors based on a more complex fundamental function which contains more detailed information about phase connectedness and hence certain information about long-range orders (Zeman, 2003).

The fundamental function can defined as

$$\lambda_r(\mathbf{x}_1, \mathbf{x}_2, \alpha) = \begin{cases} 1, & \text{if } \mathbf{x}_1\mathbf{x}_2 \subset D_r(\alpha), \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

i.e., a function which equals to 1 when the segment $\mathbf{x}_1\mathbf{x}_2$ is contained in the phase r for the sample α and zero otherwise. The lineal path function, denoting the probability that the $\mathbf{x}_1\mathbf{x}_2$ segment lies in the phase r, then follows directly from the ensemble averaging of this function

$$L_r(\mathbf{x}_1, \mathbf{x}_2) = \overline{\lambda_r(\mathbf{x}_1, \mathbf{x}_2, \alpha)}. \tag{2}$$

Under the assumptions of statistical homogeneity and isotropy, the function simplifies to

$$L_r(\mathbf{x}_1, \mathbf{x}_2) = L_r(\mathbf{x}_1 - \mathbf{x}_2) = L_r(\|\mathbf{x}_1 - \mathbf{x}_2\|). \tag{3}$$

Obviously, if the points $\mathbf{x}_1$ and $\mathbf{x}_2$ coincide, the lineal path function takes the value of volume fraction of the phase r. On the other hand, for points $\mathbf{x}_1$ and $\mathbf{x}_2$ that are far apart the lineal path function vanishes.

3. Algorithmic formulation

The generation of SEPUC is usually based on digital images, which are discretized representation of a studied medium. The segments are then defined as a set of pixels connecting two pixels $\mathbf{p}_1$ and $\mathbf{p}_2$ with the coordinates within the image $\mathbf{p}_i = (w, h)$, $w \in (1, W)$ and $h \in (1, H)$, where W and H are the dimensions of the image (see Fig. 2). The sets of pixels for segments starting in $\mathbf{p}_1 = (1,1)$ and ending in $\mathbf{p}_2 = (w, h)$ are obtained by algorithm given in (Bresenham, 1965). The group of segments is complemented by the ones starting in $\mathbf{p}_1 = (1, H)$ and ending in $\mathbf{p}_2 = (w, h)$ to cover all possible lengths and orientations within the image. Once having the defined segments, the computation of lineal path function involve simple translations of each segment throughout the image and the comparison whether all pixels of the segment at a given position correspond to image pixels with the value representing the investigated phase.

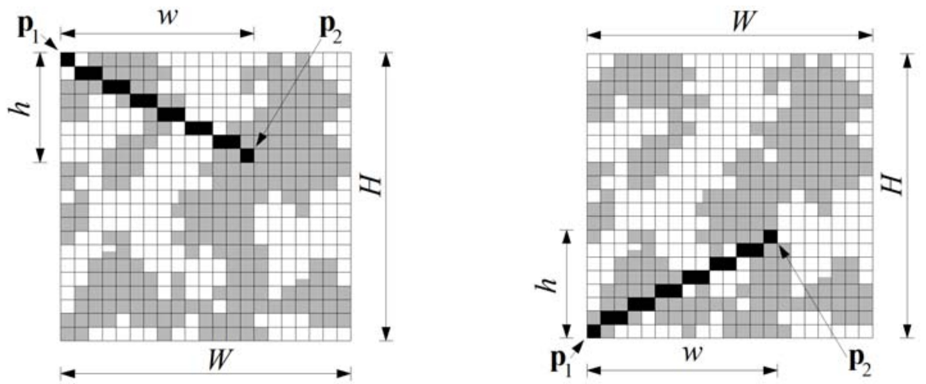


Fig. 2: Schema of the lineal path function.

Since the generation of segments can be done only once for a given image size, this part of the code does not necessarily need to be so fast. The crucial part of the code is the translation of the segment and the comparison with the image, which is called repeatedly to estimate the descriptor. Having a single CPU, the translations and comparisons needs to be performed sequentially, see Fig. 3.

```
Generate_Segments();
 for( i=0; i < nsegments; i++ ){
    for( j=0; j < npixels; j++ ){
       Is_Inside();
    }
 }
```

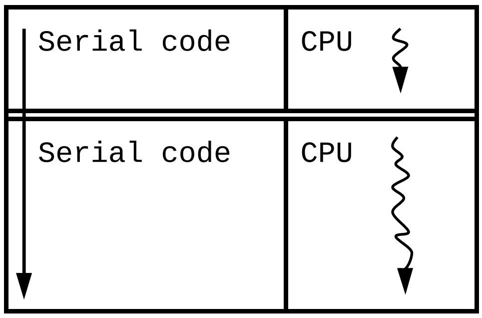


Fig. 3: Schema of the sequential code.

Last years witnessed increasing popularity of parallel computations on GPUs. The reason is the high performance at relatively low cost. Moreover, the CUDA simplifies the GPU-based software development by using the standard C language, see (NVIDIA Corporation, www). We used the high number of simple GPU threads to compute the translations and comparisons of segments simultaneously, see Fig. 4.

```
Generate_Segments();


 // Parallel kernel
 Is_Inside<<< >>>(nsegments,npixels);

```

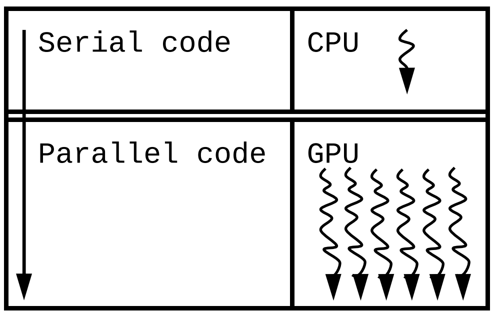
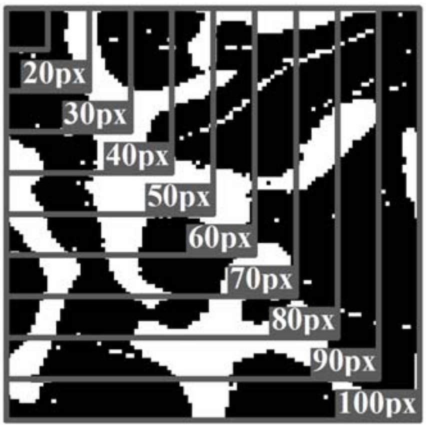


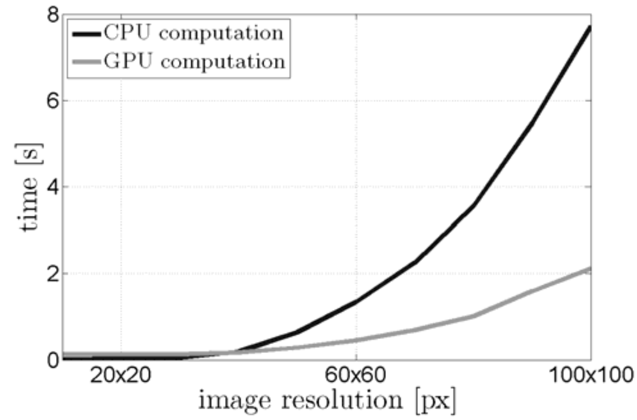
Fig. 4: Schema of the parallel code.

4. Results and conclusions

We have compared the sequential variant of lineal path function calculation on a single CPU with the parallel one using the GPU. The particular computations were performed on INTEL Core 2 Duo CPU E 6750 @ 2.66 GHz, 3.25 GB RAM, GPU - GeForce 8600 GTS with Microsoft Windows XP Professional SP 3 operating system and the CUDA v. 1.1 compute capability. The efficiency of GPU parallelism was demonstrated on evaluation of lineal path function for 10 two-dimensional images with the size varying from 10×10 px to 100×100 px, see Figure 5a. Figure 5b shows the amount of time necessary for one evaluation of lineal path function depending on the image size. One can see that for very small images, the usage of CPU outperforms the GPU because of additional time spent by copying the data from main memory RAM to GPU memory. Nevertheless, the parallelism of GPU gains for images larger than 50×50 px and the time savings increase exponentially.



(a)



(b)

Fig. 5: (a) Testing images; (b) Comparison of CPU and GPU performance.

Acknowledgement

The financial support of this work by the Czech Science Foundation (projects No. 105/11/0411 and 105/11/P370) is gratefully acknowledged.

References

Bresenham, J.E. (1965) Algorithm for computer control of a digital plotter. IBM System journal, 4, 1, pp. 25-30.

Jiroušek, O., Vavřík, D., Jakůbek, J. & Dammer, J. (2008) Correlation of trabecular bone mechanical properties to its microstructure using μCT-based FE modeling, in: Proc. Engineering Mechanics, Institute of Thermomechanics AS CR, Prague.

Lu, B. & Torquato, S. (1992) Lineal-path function for random heterogeneous materials, Physical Review E, 45, 2, pp. 922-929.

NVIDIA Corporation (www) 2701 San Tomas Expressway Santa Clara, CA 95050, USA. http://developer.nvidia.com/page/home.html

Torquato, S. (2002) Random heterogenous materials. Springer-Verlag, New York.

Vorel, J. (2009) Multi-scale modeling of composite materials. Ph.D. thesis, CTU in Prague, Prague.

Vorel, J. & Šejnoha, M. (2009) Evaluation of homogenized thermal conductivities of imperfect carbon-carbon textile composites using the Mori-Tanaka method. Structural Engineering and Mechanics, 33, 4, pp. 429-446.

Vorel, J., Zeman, J., Šejnoha, M. & Tomková, B. (2011) Homogenization of plain weave composites with imperfect microstructure: Part II-Analysis of real-world materials, submitted for publication.

Zeman, J. (2003) Analysis of Composite Materials with Random Microstructure. Ph.D. thesis, CTU Reports, 7, 3 CTU Press, Prague.

Zeman, J., Šejnoha, M. (2007) From random microstructures to representative volume elements. Modelling and Simulation in Materials Science and Engineering, 15, 4, S325-S335.