DOI: https://doi.org/10.14256/JCE.2135.2017

Primljen / Received: 12.6.2017.
Ispravljen / Corrected: 17.10.2017.
Prihvaćen / Accepted: 6.11.2017.
Dostupno online / Available online: 10.1.2018.

GPU based parallel FDEM for analysis of cable structures

Authors:

Milko Batinić, MCE
University of Split
Faculty of Civil Engineering, Arch. and Geodesy
milko.batinic@gradst.hr

Assist.Prof. Hrvoje Smoljanović, PhD. CE
University of Split
Faculty of Civil Engineering, Arch. and Geodesy
hrvoje.smoljanovic@gradst.hr

Prof. Ante Munjiza, PhD. CE
University of Split
Faculty of Civil Engineering, Arch. and Geodesy
ante.munjiza@gradst.hr

Prof. Ante Mihanović, PhD. CE
University of Split
Faculty of Civil Engineering, Arch. and Geodesy
ante.mihanovic@gradst.hr

Preliminary report

Milko Batinić, Hrvoje Smoljanović, Ante Munjiza, Ante Mihanović

GPU based parallel FDEM for analysis of cable structures

A combined finite-discrete element method, adapted for the analysis of a parallel cable element model using graphic cards, is presented in the paper. The basic objective is to speed up sequential computation time by one or two orders of magnitude. The developed solution is implemented in the open-source FDEM Y code. Performance measurements for this solution are conducted on simple examples, and relevant discussions are made.

Key words:
finite-discrete element method, cable element, GPU parallelism

Prethodno priopćenje

Milko Batinić, Hrvoje Smoljanović, Ante Munjiza, Ante Mihanović

Paralelni model kabela za grafičke kartice baziran na metodi konačno diskretnih elemenata

U radu je prikazana adaptacija kombinirane metode konačno diskretnih elemenata za paralelni model kabelskog elementa pomoću grafičkih kartica. Osnovni cilj je ubrzanje proračunskog vremena serijskog koda za jedan ili dva reda veličine. Razvijeno rješenje je implementirano unutar slobodno dostupnog FDEM Y koda. Mjerenje performansi razvijenog rješenja kao i diskusija o tome prikazani su na jednostavnim primjerima.

Ključne riječi:
metoda konačno diskretnih elemenata, kabelski element, GPU paralelnost

Vorherige Mitteilung

Milko Batinić, Hrvoje Smoljanović, Ante Munjiza, Ante Mihanović

Parallelmodell von Kabeln für Grafikkarten, basiert auf der Methode der finiten diskreten Elemente

In der Arbeit wird die Anpassung der kombinierten Methode der finiten diskreten Elemente für ein paralleles Kabelmodell anhand von Grafikkarten dargestellt. Der Schwerpunkt lag dabei auf der Beschleunigung der Berechnungszeit des Seriencodes um eine oder zwei Größenordnungen. Die entwickelte Lösung wurde innerhalb der frei zugänglichen FDEM Y Codes integriert. An einfachen Beispielen wurden die Messung der Leistungen der entwickelten Lösung sowie entsprechende Diskussionen dargestellt.

Schlüsselwörter:
Methode der finiten diskreten Elemente, Kabelelement, GPU-Parallelität

1. Introduction

Cables are very common structural elements that are used in forming three-dimensional structures such as suspended bridges, transmission lines, cable transportation systems, mooring systems, etc. They transfer load exclusively by axial force, and can therefore be considered as simple carrying structural systems. On the other hand, significant difficulties are encountered in computational modelling of cables due to their highly non-linear behaviour. Many numerical models for cables exhibit significant numerical instability or poor efficiency. Simplified solutions provide overly inaccurate solutions that even fail to accurately describe equilibrium in the final deformed configuration of the cable.

1.1. State of the art

The analysis of cable elements has been in the focus of interest for many years. In 1691 the Bernoulli brothers, Leibnitz and Huygens developed equilibrium equations for inextensible cables suspended from their ends and subjected to gravity loads. Leibnitz used the theory of infinitesimal calculation to derive the equation of this curve. In 1891, Routh solved the equation for symmetrically suspended cables consisting of a linear elastic material, while in 1981 Irvine adopted the Lagrangian approach to the solution of non-symmetrical suspended elastic cables by formulating an expression for the tangent stiffness matrix. It should be noted that the above-mentioned approaches are based on the hypothesis of small deformations, meaning that the forces are integrated with respect to the initial configuration of the cable. Also, the analytical solution of the cable structure is known for only a limited number of load cases and boundary conditions.

For this reason, cables are most frequently analysed by numerical methods in which the cable structure is discretized into smaller elements. Simple equations that model these smaller elements are then assembled into a larger system of equations that models the entire problem. Two distinct methods can be differentiated among various numerical methods that have been proposed for cable structure modelling [1].

In the first approach, which is regarded as standard in the finite element method, polynomial functions are used as basic functions to describe the shape and displacement within cable elements. The simplest finite elements in this approach are two-node straight elements [2]. These elements exhibit axial stiffness only, and are generally applicable to cables with a small curvature gradient, as in pre-stressed cables. In the case of large-curvature loose cables, geometry is described by a large number of finite elements; thus, the analysis becomes inefficient due to a large number of degrees of freedom.

Another numerical model in this approach is the multi-node isoperimetric element, which is achieved by adding more nodes to the finite element. The elements are usually three- or four-noded, depending on whether parabolic or cubic interpolation functions are used. Rotational degrees of freedom are usually added to the nodes in order to ensure continuity in curvature between the two finite elements [3]. This type of finite element describes much better the geometry of the cable, although a large number of finite elements is also necessary for large displacement and curvature gradient.

The second approach uses analytical solution to describe the shape and displacement within cable elements, taking into account the type of load applied along the cable [4–9]. This group of numerical models includes parabolic elastic elements that are widely applied due to their simplicity, and an extension of Irvine model developed, inter alia, by Ahmadi-Kashani and Bell [10]. The main feature of these numerical models is that, for the type of load from which their basic functions have been derived, they show correspondence with analytical solution with very few or even only one final element, while requiring a much larger number of finite elements for a different type of load.

The combined finite discrete element method (FDEM) proposed by Munjiza [11–13] constitutes one of the methods that are currently widely applied in the analysis of engineering structures.

1.2. FDEM numerical algorithm

The combined finite-discrete element method (FDEM), presented by Munjiza [11–13], merges finite element tools and technics with discrete element algorithms. Finite element-based analysis of continua is merged with discrete element-based transient dynamics, contact detection, and contact interaction solutions. The FDEM was mainly developed for simulation of fracturing problems involving a large number (from a few thousands to more than a million) deformable bodies that interact with each other and may split and separate during the analysis. In the framework of this method, the deformable bodies (discrete elements) are discretized by triangular (2D) or tetrahedral (3D) finite elements. The material non-linearity is considered via the material law at Gauss points, while the fracture and fragmentation of discrete elements are considered through the displacement-based contact elements that are implemented within a finite element mesh. The interaction between discrete elements is considered through the contact interaction algorithm for normal forces, which is based on potential contact forces. The method relies on an explicit numerical integration of the equation of motion. All previously mentioned numerical algorithms are implemented in the open source Y2D (for 2D analysis) and Y3D (for 3D analysis) numerical code.

The FDEM has found wide application in various fields of science such as structural analysis [14–16], rock mechanics [17, 18], maritime engineering [19], biomedical engineering [20] and mechanical engineering [21].

Sequential CPU algorithms for FDEM problems have been developed, including Munjiza-NBS [22] and MR [23] algorithm for contact detection, combined single and a smeared crack model for fracture and fragmentation [24], penalty function method based potential contact force for contact interaction [25], and time saving algorithms for force evaluation [11]. All above mentioned algorithms are part of the open source FDEM Y2D and Y3D packages for the analysis of two dimensional problems and three dimensional problems, respectively.

The FDEM is calculation intensive and, consequently, it is difficult to analyse large scale problems using the sequential CPU hardware. Different types of parallelisation solutions have therefore been explored, including the shared memory parallel computers [26, 27], distributed memory parallel computers [28], hardware independent virtual parallel machine framework for FDEM [29], and the MPI static [30] and dynamic space decomposition [31].

Parallelization procedures utilize hardware in similar fashion: concurrent job execution on many processor cores working on a specific part of the domain with communication in-between. The usage of graphics processing units (GPUs) for both the DEM [32] and 2D coupled FEM/DEM [33] analysis has been explored. The GPU parallelization of the coupled FEM/DEM approach (CDEM) was described by Wang et al. [33]. However the parallelization of FDEM problems using the GPU itself has been explored to a somewhat lesser extent.

1.3. GPU hardware and software model

The Graphic (Visual) Processing Unit – GPU [34], is a type of hardware that was initially dedicated for the creation of computer images, but has become a modern-day supplement for high end processing CPUs. Although the term GPU was coined by NVidia in 1999 for its GTS 256 model [34], it now generally includes all historical image creation hardware solutions from the 80s up to the present time.

Depending on the manufacturer and specific architecture generation, common basic building blocks (BBB) of GPU's can be listed as follows:

- Single precision processor (SPP) – consists of floating-point unit (FPU) and arithmetic- logic unit (ALU), depicted in Figure 1 – used for simple algebraic operations
- DPP - Double precision processors; 64-bit floating point operands
- SFU - Special functions processor - single precision mathematical transcendental functions – *sin*(), *cos*(), *log*, *exp*. etc.)
- LD/ST – load and store units

Different number of specific BBBs constitute streaming multiprocessors (SMM) (see Figure 1), which are clusters of processors that share parts of the chip memory. The chip is made of 16 streaming multiprocessors, each made of four quadrants with 32 SPPs, 8 LD/STs, 8 SFUs, and 2 DPP cores.

The GPU memory (hardware parameters for Maxwell GTX 980 [35]) is divided into:

- Registers
- Unified L1/Texture cache
- Shared memory
- Local memory
- Read only cache
- Global memory

where registers are visible only to processors currently using it, shared memory is shared among processors within a multiprocessor, and global memory is shared among all processors. The code that is run in parallel on the GPU processors is called a kernel. Each copy executed on one of the processors is called a thread. It is an exact copy of the kernel, running concurrently with other copies – threads.

To make full use of the data parallelism, threads need to have an aligned memory access, and are therefore grouped into warps. Thirty-two threads constitute one warp. As managing all execution related operations manually can be quite tedious, different types of application programming interfaces (API) have been developed. The proprietary NVidia software is used, although open source solutions are also available (OpenCL).

The FDEM based GPU parallel algorithm for the analysis of cable structures is presented in the paper. The algorithm exploits massive parallelism offered by modern day general computation graphics cards, as well as advantages gained by combining discrete element techniques with the FEM.

All computations presented in this paper were performed using the NVidia GTX 980 card - Maxwell architecture (hardware parameters depicted in Figure 1). To the authors' knowledge, no similar codes for parallel processing of cable structures have so far been developed.

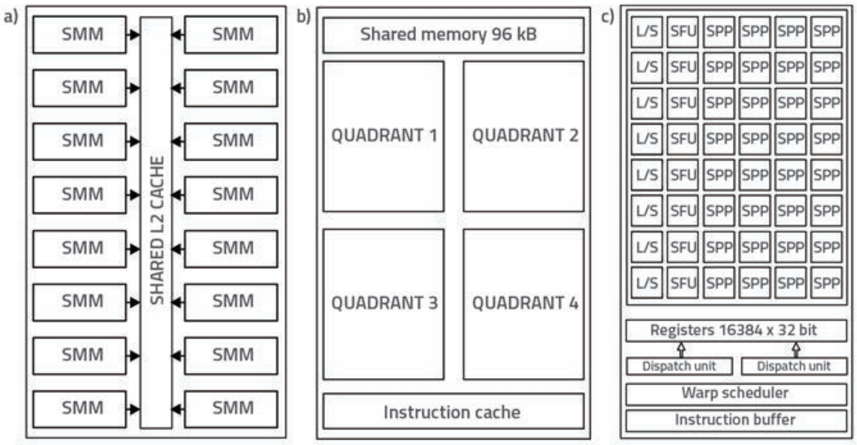

Figure 1. Maxwell GTX 980 –a) chip layout b) SMM layout c) one SMM quadrant layout, as per [35]

2. GPU algorithm

The numerical algorithm for the analysis of cable structures [36] has been adapted in this section for parallel processing using GPUs. The discretisation of structure, and detailed information related to axial carrying mechanism and time integration of the equation of motion, are shown below.

2.1. Discretisation

In this algorithm, the cable is discretised with two-noded finite elements that can transfer axial forces only, i.e. the forces in the direction of their axes [36]. Masses are lumped into the nodes of the finite elements as shown in Figure 2.

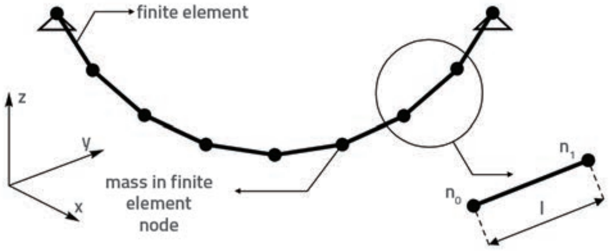

Figure 2. Discretisation of cable structure

2.2. Axial carrying mechanism

Two-noded finite elements support axial stresses only, i.e. the stresses in the direction of their axes [36]. The geometry of the finite element is defined by two nodes, as shown in Figure 3. Each node is described by its global initial and current Cartesian coordinates (x,y,z).

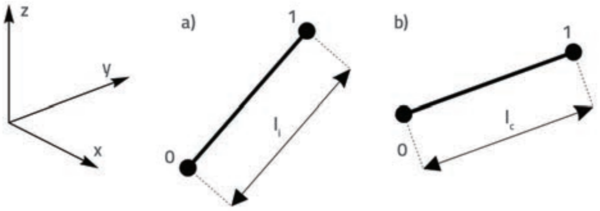

Figure 3. Initial and current coordinates of the nodes of the finite element.

The initial l_i and current l_c length of the finite element can be determined based on known coordinates of the nodes in the initial and current configurations, as shown in Figure 3. The axial strain in each finite element is obtained according to

$$\varepsilon = (l_c - l_i) / l_i \tag{1}$$

By using the linear viscoelastic material behaviour, the corresponding stresses are obtained according to

$$\sigma = E\varepsilon + \bar{\mu}\dot{\varepsilon} \tag{2}$$

where E represents the modulus of elasticity, $\bar{\mu}$ represents the damping coefficient and $\dot{\varepsilon}$ stands for the velocity of the change in strain.

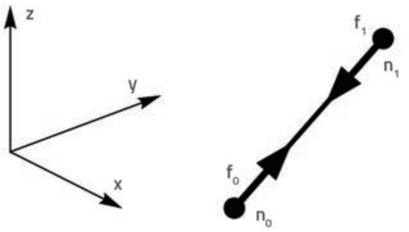

Figure 4. Equivalent nodal forces due to axial stress of finite element

Equivalent nodal forces in the direction of the finite element axis (Figure 4) are obtained according to

$$f_0 = f_1 = \sigma A \tag{3}$$

where A represents the cross-sectional area of the cable

2.3. Time integration of the equation of motion

The shape of a cable structure and its position in space at any given time is defined by current coordinates of the finite element nodes x_i, where i is associated with the degree of freedom. Each node has three degrees of freedom, which relates to translation in the x, y and z directions. Similarly, the velocity field and acceleration field are defined by nodal velocities v_i and nodal accelerations a_i, respectively [11-13].

The explicit time integration scheme is applied to each node and each degree of freedom. Nodal forces from the axial carrying mechanism and external loads, such as gravity load or some other external load, are all added together and the total nodal force f_i, associated with each degree of freedom, is obtained. The dynamic equilibrium for each degree of freedom is therefore given by

$$m_i a_i = f_i \tag{4}$$

where m_i is the mass of the corresponding node.

A central difference time integration scheme, based on explicit integration of the governing equation for each degree of freedom, is used for integration of the above equation. The scheme can be formulated as follows

$$\begin{aligned} v_{i,t+\Delta t/2} &= v_{i,t+\Delta t/2} + \Delta t\, f_{i,t} / m_i \\ x_{i,t+\Delta t} &= x_{i,t} + \Delta t\, v_{i,t+\Delta t/2} \end{aligned} \tag{5}$$

where Δt is a time step.

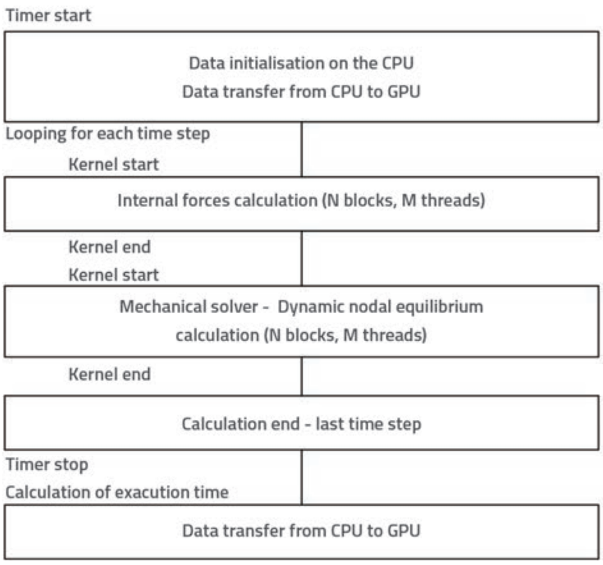
Timer start
Data initialisation on the CPU
Data transfer from CPU to GPU
Looping for each time step
Kernel start
Internal forces calculation (N blocks, M threads)
Kernel end
Kernel start
Mechanical solver - Dynamic nodal equilibrium calculation (N blocks, M threads)
Kernel end
Calculation end - last time step
Timer stop
Calculation of exacution time
Data transfer from CPU to GPU

Figure 5. Flow chart for proposed parallelisation algorithm of cable element

Considering the overall nature of parallel algorithms, models with high number of elements tend to perform better than smaller models. Therefore, the adopted discretisation with simplest two-noded finite elements is considered to be efficient for arbitrary geometry.

The schematic flowchart describing full numerical procedure is shown in Figure 4. It is worth noting that the proposed numerical procedure does not require assembly of either stiffness or mass matrices, which makes it suitable for parallel programming.

From the standpoint of programming, the proposed cable element formulation data is stored into 1D arrays and the *float* data structure type is used (comparison with *double* is given). Each 1D vector is aligned to 384 bit strides so as to ensure an optimum global memory access times. The data shared among all kernels is stored in shared memory, while all other instances of shared data are stored in registers to improve performance. Local temporary variables are introduced to reduce the overall register count per thread. These variables are used for multiple global variable calculations. The initial system configuration is calculated at the CPU (element length, nodal connectivity), and the data is transferred from the CPU (host) to the GPU (device). After completion of all calculations for all time steps, the results are copied back from the GPU to the CPU and written into output files.

3. Numerical results

3.1. Example 1 – Verification of algorithm

Figure 5 shows initial configuration of a simple catenary system, with a fixed support in point A, and sliding support in point B, where horizontal force F=60000 N is applied. The total length of the system is l=1000 m. The calculation is done for the catenary mass *m*=5kg/m'. cross section area A = 25 cm^2, modulus of elasticity E=210 000 MPa, and Poisson coefficient n = 0.

The analytical solution for vertical deflection in point X is obtained when the system is in equilibrium, as shown in (6)

$$y = \frac{1}{H}\int_0^x\int_0^x q\,dx\,dx + C_1 x + C_2 \qquad (6)$$

The analytical solution for vertical deflection in point X is y_x=-100 m for the defined properties and boundary conditions (Figure 6).

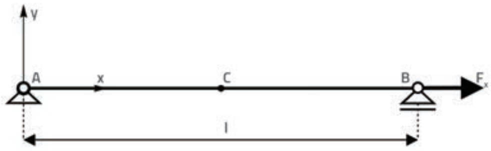

Figure 6. Initial system configuration

Comparison of vertical deflection of point X for analytical solution, based on the commercial software package SCIA Engineer [37], and the proposed parallel algorithm with different sized meshes are given in Table 1. Although the analytical solution of vertical deflection in point X does not include cable extension, material chrematistics are chosen in such a way that the achieved extension is negligible (order of magnitude: one mm).

The comparison of execution times for sequential CPU implementation of the Y-code with PC configuration (Intel core i7 processor, overclocked at 3,75 GHz, 16 GB RAM, PCI Express 2.0) and the parallel GPU code for different sized meshes, is shown in Figure 6. The GPU parallel code performs poorly for small sized problems, due to GPU nature (multiple kernel invocation overhead on small sample size), but it gradually outperforms serial code up to 80 times for systems with 10 million elements. According to Table 1, the relative error of results obtained with the commercial package SCIA Engineer is smaller compared to the FDEM. However, advantages of the proposed numerical model include geometric nonlinearity (large rotations and large displacements), while the linear material model used can easily be expanded to include different types of material nonlinearities. The contact interaction and the possibility of cable fracture can also be introduced.

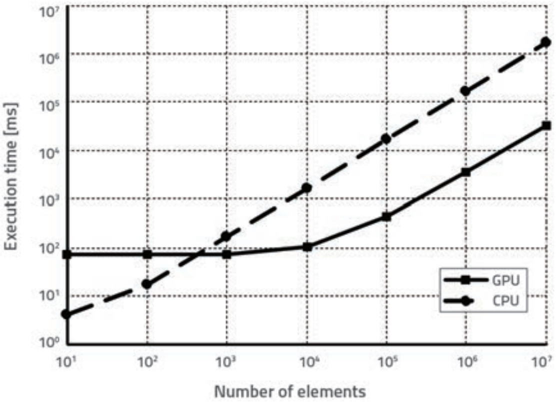

Figure 7. Comparison of execution time for serial CPU, and parallel GPU algorithm

Table 1. Vertical displacement of point C – comparison of results and relative error

Solution	Vertical deflection in point C [m]	Absolute value of relative error [%]
Analytical solution per (6)	100	0
Commercial FEM package [37] – 1000 elements	100.73	0.73
Algorithm - 100 elements	96.24	3.76
Algorithm - 1000 elements	98.27	1.73
Algorithm - 10000 elements	99.96	0.04
Algorithm - 100000 elements	99.99	0.01

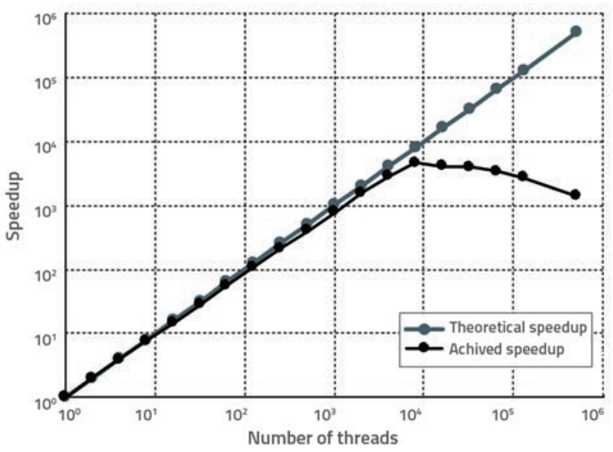

Figure 8. Parallel algorithm execution time for variable number of blocks, and threads per block

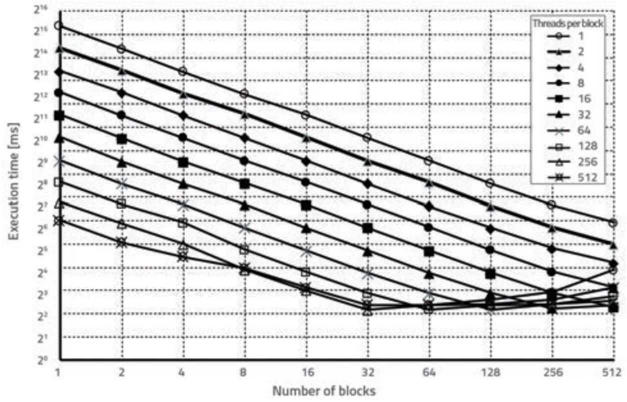

Figure 9. Relative speedup of code execution time, in comparison to running only one thread of parallel code on GPU

Considering the programming strategy applied, an optimum performance in terms of hardware utilization can be achieved by using 256 blocks, and 64 threads per block, as shown in Figure 8. The overall speedup is limited by resources used per one thread, and the maximum speedup is achieved when 16384 threads are used (Figure 9).

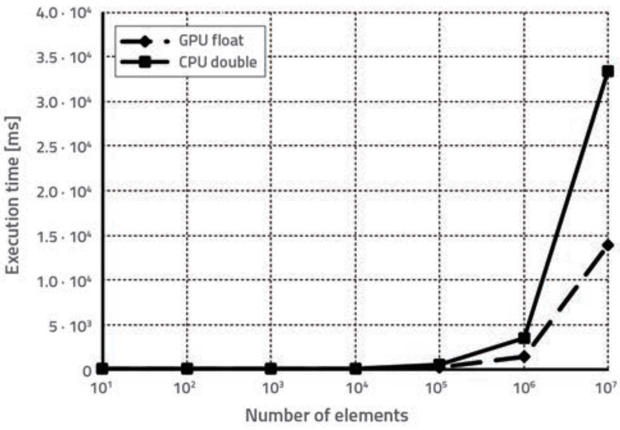

Figure 10. Execution time for parallel algorithm code using data type float vs double

Comparison of execution times for different data types is depicted in Figure 10. Due to the number of dedicated cores on the device, the overall speed execution of algorithm utilising *float* data type is up to three times faster.

3.2. Wave propagation – dynamic analysis

This example is used to demonstrate dynamic capabilities of the developed algorithm. The point A of the system depicted in Figure 10 is exposed to the oscillating speed of v_y as per equation

$$\nu_y = 2\cos(\omega t)$$

for the time interval t = 2,0 s, where ω = 0,5 π.

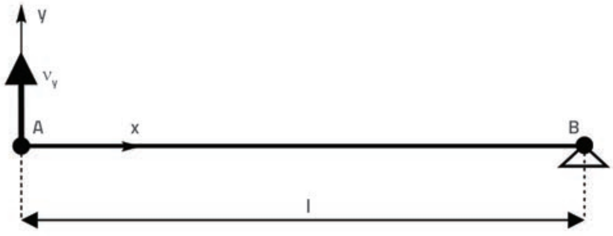

Figure 11. Wave propagation - Initial system configuration

The calculation is conducted for the catenary mass *m* =1kg/m'. cross section area A = 25 cm², modulus of elasticity E= 210 000 MPa, and Poisson coefficient ν = 0.

The system length is *l* =100.0 m, whereas the discretisation length of one element is 0.01m, for the overall number of 10000 finite elements. One million time steps are executed, each at 10^{-5} s, for the total simulation time of *t* =10 s.

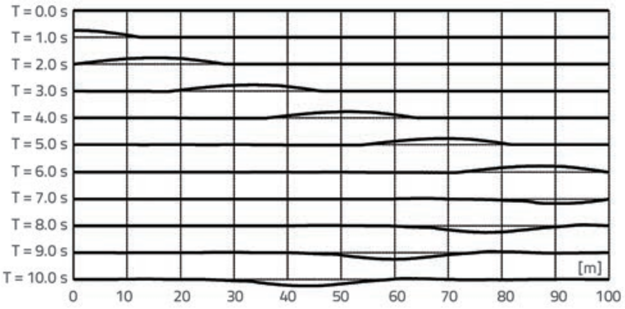

Figure 12. Wave propagation in catenary

The displacement of catenary in global Y direction is shown in Figure 12 for different time frames. A wave traveling at the speed of 16.6 m/s is induced by the oscillating motion of point A and, at time t = 6 s, it hits its fixed support in point B, and from there it travels back to point A.

3.3. Mooring - static cable analysis

Example 3 demonstrates form finding for static equilibrium of the mooring system. The mooring system depicted in Figure 12 is fixed at point A, while point B is free. At point B, the horizontal force F_x = 1000 kN, and the vertical force F_y = 600 kN, are applied to initial configuration.

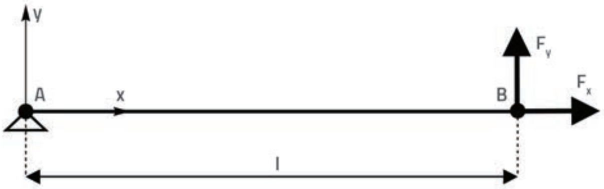

Figure 13. Mooring - Initial configuration of system

Properties for catenary are: mass *m*=5kg/m', cross section area A = 25 cm^2, modulus of elasticity E= 210 000 MPa, Poisson coefficient ν = 0, system length *l* = 500.0 m, and the discretisation length of one element is 0.01m, for the overall number of 50000 finite elements. Twenty million time steps are executed, each 5·10^{-6} s, for the total simulation time of *t* = 10 s.

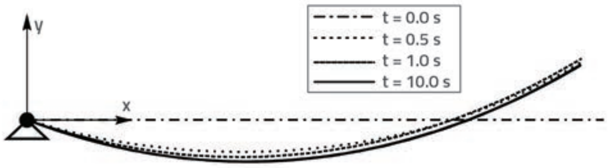

Figure 14. Equilibrium state of mooring system

Table 2. Coordinates of point B for various simulation times

Time [s]	**Global X coordinate** [m]	**Global Y coordinate** [m]
0	500.00	0.00
0.5	478.57	50.57
1	476.08	48.30
2	475.51	47.13
5	475.49	46.42
10	475.49	46.36

The system position at various time steps is shown in Figure 14. The algorithm convergence to stationary position of the system, for the simulation time of 1.0 s (two million time steps), is within 1 % for X direction, and 5 % for Y direction. The system convergence per specific simulation time is given in Table 2.

4. Conclusion

This paper presents a FDEM based GPU parallel algorithm for the static and dynamic analysis of cable structures. The algorithm is based on the combined finite-discrete element method model, using the two-noded rotation free finite elements. The information related to the parallelisation strategy and GPU implementation is briefly presented in the paper. The proposed algorithms are incorporated into the existing open-source "YFDEM" package. The model performance is demonstrated using simple benchmark tests by comparing the results obtained by the proposed algorithm with known analytical and numerical results**.**

The developed algorithm performs poorly for small systems consisting of several hundred elements, when compared to serial implementation. Its performance gradually improves and achieves the speedup of 80 times for systems with several million elements.

The proposed parallel model performs very well for systems requiring a large number of elements, and can be applied in cases when large interconnected systems of cables are used. Additional dynamic properties shown in examples 2 and 3 make it suitable for specific engineering problems, including propagation effects at large linear distribution networks, mooring of offshore structures, and analysis of large cable constructions.

In addition, the fracture, material nonlinearity, contact detection, and interaction within cable elements and other types of FDEM elements, can easily be introduced, thus providing a simulation tool for a wide spectrum of research fields, including various areas of civil engineering, maritime engineering, mechanical engineering, etc.

REFERENCES

[1] Coarita, E., Flores, L.: Nonlinear Analysis of Structures Cable - Truss, International Journal of Engineering and Technology 7 (2015) 3, pp. 160-169.

[2] Tibert, G.: Numerical analyses of cable roof structures, KTH, Structural Engineering, Superseded Departments, KTH, 1999.

[3] Thai, H.T., Kim, S.E.: Nonlinear static and dynamic analysis of cable structures, Finite Elements in Analysis and Design, 47 (2011) 3, pp. 237-246.

[4] Pipkin, A.C.: Catenary deformations of inextensible networks, Journal of Engineering Mathematics, 28 (1994) 5, pp. 401-406.

[5] Suzuki, Y., Miyata, E.S., Iverson, S.C.: Static analyses of the triangular running skyline system: A three-dimensionally movable logging cable system, Computers & Structures, 60 (1996) 4, pp. 579-592.

[6] Huang, Y., Lan, W.: Static analysis of cable structure, Applied Mathematics and Mechanics, 27 (2006) 10, pp. 1425-1430.

[7] Andreu, A., Gil, L., Roca, P.: A new deformable catenary element for the analysis of cable net structures, Computers and Structures, 84 (2006) 29-30, pp. 1882-1890.

[8] Kiureghian, A., Sackman, J.L.: Tangent Geometric Stiffness of Inclined Cables Under Self-Weight, Journal of Structural Engineering, 131 (2005) 6, pp. 941-945.

[9] Lopez-Garcia, O., Carnicero, A., Torres, V.: Computation of the initial equilibrium of railway overheads based on the catenary equation, Engineering Structures, 28 (2006) 10, pp. 1387-1394.

[10] Cuomo, M., Greco, L.: A finite element cable for the analysis of cable nets, Meccanica strutture, pp. 1-11, 2009

[11] Munjiza, A.: The combined Finite-Discrete Element Method, John Wiley & Sons, 2004. UK, https://doi.org/10.1002/0470020180

[12] Munjiza, A., Knight, E. E., Rougier, E.: Computational Mechanics Of Discontinua. John Wiley & Sons, 2011. UK, https://doi.org/10.1002/9781119971160

[13] Munjiza, A., Rougier, E., Knight, E. E.: Large strain Finite element method a practical course, John Wiley & Sons, 2015. UK

[14] Uzelac, I., Smoljanović, H., Peroš, B.: A computationally efficient numerical model for a dynamic analysis of thin plates based on the combined finite-discrete element method, Engineering Structures, 101 (2015) pp. 509-517, https://doi.org/10.1016/j.engstruct.2015.07.054

[15] Zivaljic, N., Smoljanovic, H.,Nikolic, Z.: A combined finite-discrete element model for RC structures under dynamic loading, Engineering Computations, 30 (2013) 7, pp. 982-1010.

[16] Smoljanović, H., Živaljić, N., Nikolić, Ž.: A combined finite-discrete element analysis of dry stone masonry structures, Engineering Structures, 52 (2013), pp. 89-100, https://doi.org/10.1016/j.engstruct.2013.02.010

[17] Grasselli G., M. O. K. C. E.: An Example of Realistic Modelling of Rock Dynamics Problems : FEM / DEM Simulation of Dynamic Brazilian Test on Barre Granite, Rock Mechanics and Rock Engineerin,g 43 (2010) 6, pp. 707-716.

[18] Zhao, Q., Lisjak, A., Mahabadi, O., Liu, Q., Grasselli, G.: Numerical simulation of hydraulic fracturing and associated microseismicity using finite-discrete element method, Journal of Rock Mechanics and Geotechnical Engineering, 6 (2014) 6, pp. 574-581.

[19] Bai, X., Avital, E.J., Munjiza, A., Williams, J.J. R.: Numerical simulation of a marine current turbine in free surface flow, Renewable Energy, 63 (2014), pp. 715-723, https://doi.org/10.1016/j.renene.2013.09.042

[20] Xu, D., Kaliviotis, E., Munjiza, A., Avital, E., Ji, C., Williams, J.: Large scale simulation of red blood cell aggregation in shear flows, Journal of Biomechanics, 46 (2013) 11, pp. 1810-1817.

[21] Kruggel-Emden, H., Rickelt, S., Stepanek, F., Munjiza, A.: Development and testing of an interconnected multiphase CFD-model for chemical looping combustion, Chemical Engineering Science, 65 (2010) 16, pp. 4732-4745.

[22] Munjiza, A., Andrews, K.: NBS contact detection algorithm for bodies of similar size, International Journal for Numerical Methods in Engineering, 149 (1998), pp. 131-149, https://doi.org/10.1002/(SICI)1097-0207(19980915)43:1<131::AID-NME447>3.0.CO;2-S

[23] Munjiza, A., Rougier, E., John, N.W.M.: MR linear contact detection algorithm, International Journal for Numerical Methods in Engineering, 66 (2006) 1, pp. 46-71.

[24] Munjiza, A.: Combined single and smeared crack model in combined finite discrete element analysis, International Journal for Numerical Methods in Engineering, 44 (1999) 1, pp. 41-57.

[25] Munjiza, A., Andrews, K.R.F.: Penalty function method for combined fnite - discrete element systems comprising large number of separate bodies, International Journal for numerical methods in engineering, 49 (2000) , pp. 1377-1396, https://doi.org/10.1002/1097-0207(20001220)49:11<1377::AID-NME6>3.0.CO;2-B

[26] Owen, D.R.J., Feng, Y.T.: Parallelised finite/discrete element simulation of multi-fracturing solids and discrete systems, Engineering Computations, 18 (2001) 3/4, pp. 557-576.

[27] Useche, J., Albuquerque, E. L.: Transient dynamic analysis of shear deformable shallow shells using the boundary element method, Engineering Structures, 87 (2015), pp. 1-7, https://doi.org/10.1016/j.engstruct.2015.01.006

[28] Hussain, H., Ur, S., Malik, R., Hameed, A., Ullah, S., Bickler, G., Min-allah, N., Bilal, M., Zhang, L., Yongji, W., Ghani, N., Kolodziej, J., Zomaya, A.Y., Xu, C., Balaji, P., Vishnu, A., Pinel, F., Pecero, J. E., Kliazovich, D., Bouvry, P., Li, H., Wang, L., Chen, D., Rayes, A.: A survey on resource allocation in high performance distributed computing systems, PARALLEL COMPUTING, 39 (2013) 11, pp. 709-736.

[29] Lei, Z., Rougier, E., Knight, E.E., Munjiza, A.: A framework for grand scale parallelization of the combined finite discrete element method in 2D, Computational Particle Mechanics, 1 (2014) 3, pp. 307-319.

[30] Schiava D'Albano, G.G.: Computational and Algorithmic Solutions for Large Scale Combined Finite-Discrete Elements Simulations, QMUL, Dissertation, 2013. UK

[31] Lukas, T., Schiava D'Albano, G.G., Munjiza, A.: Space decomposition based parallelization solutions for the combined finite-discrete element method in 2D, Journal of Rock Mechanics and Geotechnical Engineering, 6 (2014) 6, pp. 607-615.

[32] Zhang, L., Quigley, S.F., Chan, A.H.C.: A fast scalable implementation of the two-dimensional triangular Discrete Element Method on a GPU platform, Advances in Engineering Software, 60-61 (2013), pp. 70-80, https://doi.org/10.1016/j.advengsoft.2012.10.0

[33] Wang, L., Li, S., Zhang, G., Ma, Z., Zhang, L.: A GPU-Based Parallel Procedure for Nonlinear Analysis of Complex Structures Using a Coupled FEM / DEM Approach, 2013 (2013) iii, pp. 1-16.

[34] N.G.: GeForce 256 and RIVA TNT Combiners How to best utilize the per-pixel operations using OpenGL on NVIDIA Graphics Processors, Processors, pp. 1-28, 1999.

[35] NVIDIA GeForce GTX 980, pp. 1-32, 2014. https://international.download.nvidia.com/geforce-com/international/pdfs/ GeForce_GTX_980_Whitepaper_FINAL.PDF

[36] Divić, V., Simulacije krajnjih graničnih stanja pod djelovanjem vjetra metodom konačno-diskretnih elemenata 2014., doktorska disertacija, Fakultet građevinarstva, arhitekture i geodezije, Split

[37] SCIA Engineer 2017, https://www.scia.net/ accessed 05.05.2017