Dmitry POGORELOV, Alexander RODIKOV, Roman KOVALEV*
Bryansk State Technical University, Laboratory of Computational Mechanics
bulv. 50 let Oktyabrya 7, Bryansk, 241035, Russia
Corresponding author. E-mail: kovalev@umlab.ru

PARALLEL COMPUTATIONS AND CO-SIMULATION IN UNIVERSAL MECHANISM SOFTWARE. PART II: EXAMPLES

Summary. The second part of the paper continues a discussion on the topic of parallel computations in railway dynamics. The algorithms described in the first part of the paper are applied to parallel simulation on computers with multicore processors of six different models of rail vehicles and trains with the number of degrees of freedom from about one hundred to more than 20 thousands. A considerable simulation speedup is reported. In addition, an example of evaluation of wheel profile wear on multicore processors and comparison of different approaches to multi-variant computations are considered.

1. INTRODUCTION

A large variety of models is used in the simulation of railway vehicle and train dynamics. The number of degrees of freedom and, accordingly, the level of computational complexity differ considerably depending on the problem to be solved. A single-vehicle model usually has about one hindered DOF; a 3D train model may count several thousands of DOF. Inclusion of flexible parts and rails increases the number of DOF up to dozens of thousands. A speedup factor is the main characteristic of efficiency of the parallel algorithms described in the first part of the paper. The speedup results given below can be used to compare different algorithms and their implementations.

In Section 2, we consider a practical application of parallel algorithms, described in the first part of this paper [1], and implemented in Universal Mechanism software [2].

Our goal is to show that the parallel algorithm is efficient not only for models with large numbers of DOF but also for usual models of rail vehicles such as freight, passenger cars, or a locomotive. Both rigid body and hybrid vehicle models as well as vehicle–bridge interaction examples are considered. Three rail models are used for simulation: massless rails [3], 'moving' rigid body rails [4], and flexible rails as Timoshenko beams.

Examples of a parallelism in the evaluation of wheel profile wear and in multi-variant computations are considered in Sections 3, 4.

2. EXAMPLES OF PARALLEL SIMULATION OF RAIL VEHICLE AND TRAIN DYNAMICS

In this section, we consider examples of parallel simulation of rail vehicle dynamics. Results were obtained on two computers with different processors:
- Intel R CoreTM i7 4790 CPU, 4 GHz, 8 GB RAM, four cores, and eight logical processors (Processor 1)
- AMD Ryzen 7 2700 8-Core CPU, 3.2 GHz, 32 GB RAM (Processor 2).

We consider six models of rail vehicles and trains. Some statistical data on the models are collected in Tab. 1. Numbers in brackets correspond to degrees of freedom for the models.

<div align="right">Table 1</div>

<div align="center">Model statistics: number of elements (DOF)</div>

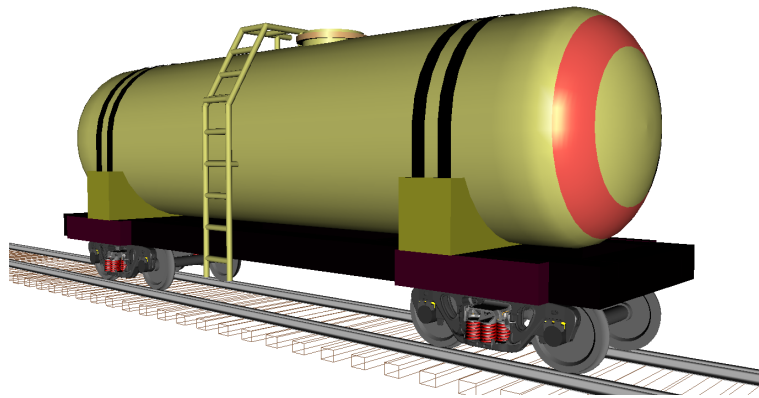|	FrCar	Diesel	LTD	Y25	PTrain	FTrain
Rigid bodies	19(114)	45(118)	52(52)	30(114)	42(252)	673(3392)
Force elements	101	186	106	119	1490	6197
Massless rail model	+	+	-	-	-	-
Moving rigid rails	-	-	-	+(16)	-	-
Flexible rails	-	-	-	-	+(7248)	+(16920)
Flexible bodies	-	-	-	1(43)	3(222)	2(144)
Total number of DOF	114	118	52	173	7722	20456

Fig. 1. Freight car with three-piece bogies

Model 1. Freight car with three-piece bogies, nickname 'FrCar', Fig. 1. A description of the model can be found in [5]. The main feature of the model consists in use of more than one hundred compliant frictional contacts between the bodies. Frictional wedges are modeled by rigid bodies with six DOF each. The model has no joints, and its simulation with the parallel algorithm described in [1] is faster using even one thread in comparison with other solvers available in UM. Simulation in our case corresponds to the vehicle motion in a tangent section of 300 m length with a constant speed of 20 m/s. The massless rail model is used [3] and the FASTSIM algorithm is applied for the computation of creep forces.
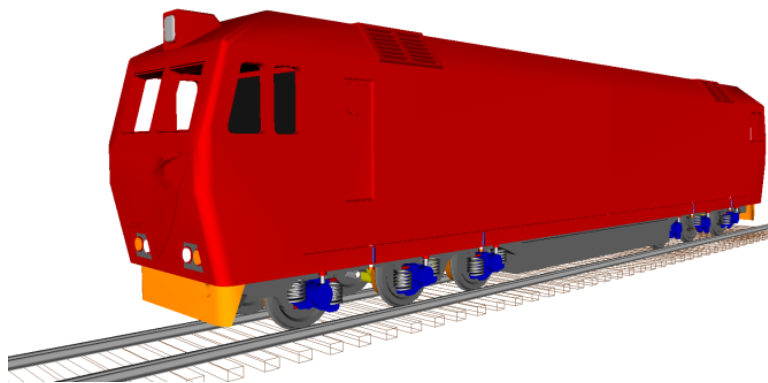

Fig. 2. Diesel-locomotive TEP70

Model 2. Diesel locomotive TEP70, nickname 'Diesel', Fig. 2. The model is included in the set of models as a typical example of a locomotive. The passenger locomotive TEP70 runs on two three-axle bogies. In contrast to the freight car models, this model contains 24 rotational joints, which are replaced by force elements according to the Cartesian formulation. In our test, the vehicle runs in a curve R = 400 m with a speed of 18 m/s. The total track length is S = 600m.

For this model, the CPU time for one integration step is approximately equal for the Cartesian formulation without parallelization and for the equations in the minimum number of coordinates.
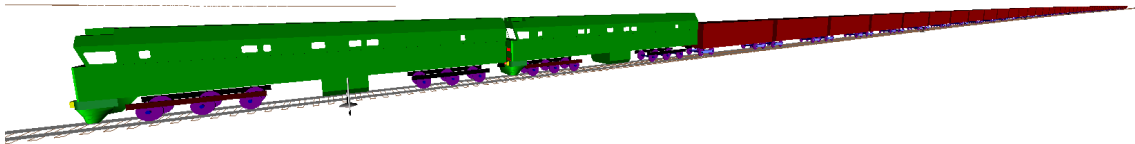

Fig. 3. 1D train for LTD benchmark, configuration 1

Model 3. Longitudinal train dynamics, 'LTD', Fig. 3. This model corresponds to an international benchmarking of longitudinal train dynamics, Configuration 1 (2 locomotives and 50 wagons) [6]. The one-dimensional (1D) train model moves with a variable speed in a track of 50 km length with a given curvature and elevation. The train operation time is 2709 s. Vehicle models have 1 DOF each. Coupling, resistance, traction, and braking forces are taken into account. With 52 DOF, it is the computationally simplest model in the tests considered in the paper.
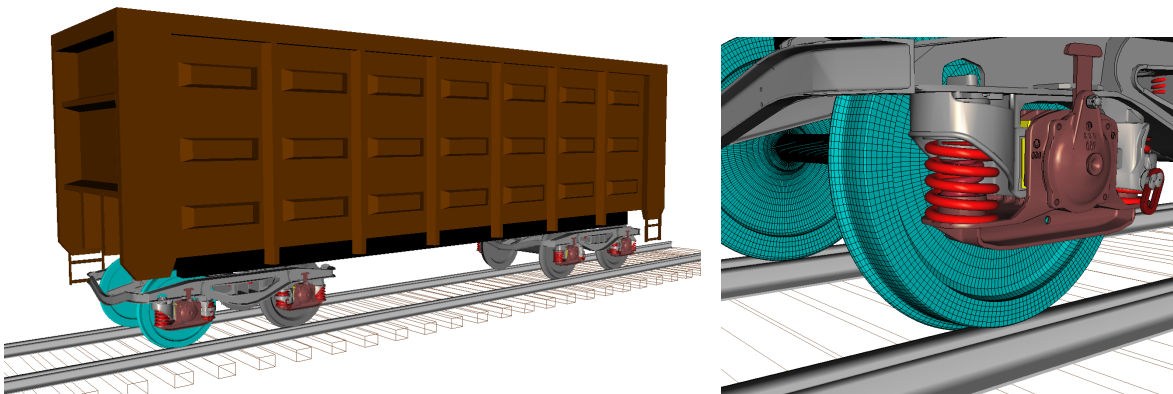

Fig. 4. Freight car with Y25 bogies; the first wheelset is flexible

Model 4. Freight car with Y25 bogies, 'Y25', Fig. 4. The model details are described in [5]. The first wheelset is modeled by a flexible body [7]. The car moves in a curve R = 300 m; the track length is 400 m; and the vehicle speed is 12 m/s. Eight 'moving rails' [4] have 2 DOF each and follow the wheelsets along the track. The model allows us to estimate the influence of the flexible wheelset on CPU costs.
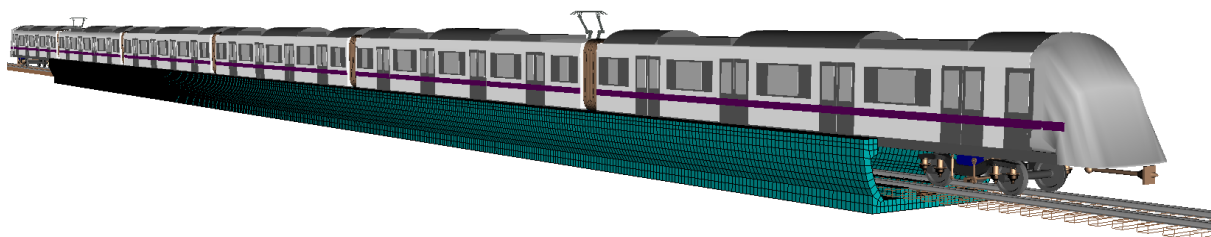

Fig. 5. 3D model of a passenger train over a flexible bridge. By courteous permission of Zhuzhou Times New Material Technology Co., Ltd and Sichuan Tongsuan Technology Co., Ltd

Model 5. Passenger train motion over a flexible bridge, 'PTrain', Fig. 5. The passenger train consists of six cars and moves over a three-section bridge of 90 m length. Rails of length 350 m are

modeled by the Timoshenko beams. The train speed is 60 km/h. This is the first of two models of three-dimensional trains moving over flexible rails and bridges considered in the paper. Flexible rails are connected with bridges by linear bushings. The total number of force elements is 1490.
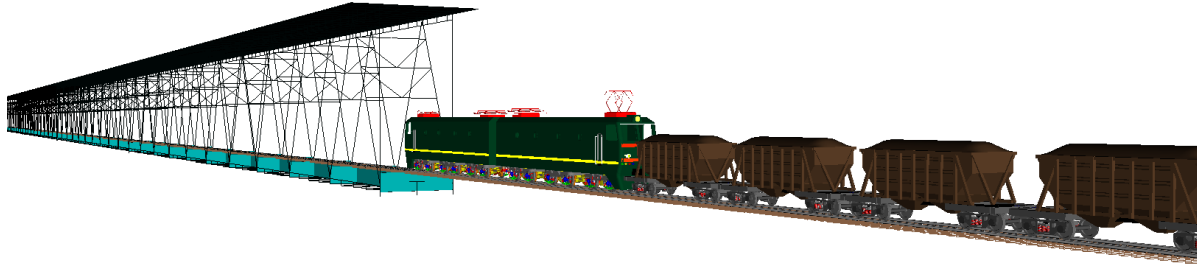

Fig. 6. 3D model of freight train and flexible bridge. By courteous permission of OAO Institut Giprostroimost.

Model 6. Freight train motion over a bridge, 'FTrain', Fig. 6. The train includes a two-section electric locomotive VL85 and 25 freight cars with three-piece bogies (see Model 1). Flexible rails are modeled by the Timoshenko beams. The train moves over a 700 m two-section flexible bridge with a speed of 100 km/h. A crosswind lateral load is taken into account. This is the biggest model in our tests with over 20 000 DOF and over 6 000 force elements.

Some information about CPU costs for the simulation of the models in one thread is presented in Tab. 2. The table includes the average values of the following variables:

h – integration step size;

T_h – CPU costs for one step size; the value does not include an overhead related to computation of simulation results, animation, plots, etc.

Other variables correspond to the CPU time for each of the parallel sections [1]:

T_{kin} – PS1 (prediction, computation of kinematics, mass matrices, internal elastic and damping forces, inertia and gravity forces);

T_{frc} – PS2 (evaluation of forces and Jacobians);

T_{addf} – PS3 (evaluation and factorization of the preconditioning matrices, computation of total forces in equations);

T_{cg_s}, T_{cg_d}, T_{cg_c}, T_{cg_t}, T_{cg_f} – PS4-8 (solving linear equations by the conjugate gradient method);

T_{fin} – PS9 (computation of new values of coordinates, velocities, and accelerations).

The CPU costs correlate with the number of DOF. The 1D LTD model with 52 DOF is the simplest one, and the 3D train model FTrain with 20456 DOF is the most expensive in the computational sense.

It is natural that computation of forces takes the major part (70-80%) of the CPU costs. Nevertheless, a speedup of all other parts of integration process increases the efficiency of parallelization.

The speedup factor for T_h versus the number of threads for simulation on a computer with a 4-core Processor 1 is shown in Fig. 7. As expected, the most efficient use of parallel computations is observed for 3D models of trains with big numbers of bodies, forces, and degrees of freedom. It is important that simulation of single rail vehicles is speeded up considerably as well.

To compare the parallel simulation with different processors, the passenger train (PTrain) model was tested on a computer with an 8-core Processor 2. Speedup factors as well as a ratio of T_h values for Processor 1 and Processor 2 are shown in Fig. 8. Simulation with Processor 1 is faster than with Processor 2 for number of threads lower 4 and slower otherwise.

3. EXAMPLE OF WEAR PREDICTION FOR THE RAILWAY WHEEL PROFILE

Let us consider the evaluation of wheel profile wear of a passenger car as an example of parallel computations on multi-core processors. The example includes three variants of a track: a tangent section of 620 m length as well as curves R = 650 m and R = 350 m of 310 m length each. Car speeds are 40, 60, and 80 km/h for the tangent and R = 650 m sections and 50 km/h in the curve R = 350 m.

Fully and half-occupied coaches were analyzed. Finally, values 0.25 and 0.4 of the friction coefficient in the wheel/rail contact were used. Consequently, the total number of simulations is 28.

Table 2

Result for step size and CPU costs (ms) by simulation in one thread

| Model	h	T_h	T_{kin}	T_{frc}	T_{addf}	T_{cg_s}	T_{cg_d}	T_{cg_c}	T_{cg_t}	T_{cg_f}	T_{fin}
FrCar	0.746	0.272	0.009	0.224 (82%)	0.013	0.008	0.011	0.012	0.001	0.001	0.002
Diesel	0.511	0.338	0.022	0.243 (72%)	0.030	0.016	0.019	0.016	0.001	0.002	0.004
LTD	8.55	0.162	0.045	0.095 (59%)	0.004	0.004	0.006	0.004	0.001	0.001	0.001
Y25	0.146	0.791	0.028	0.604 (76%)	0.040	0.018	0.047	0.078	0.005	0.004	0.004
Ptrain	0.114	11.78	0.778	8.18 (70%)	2.34	0.059	0.063	0.041	0.002	0.142	0.044
Ftrain	0.136	54.3	2.65	40.6 (75%)	8.14	0.90	0.12	0.93	0.11	0.43	0.23
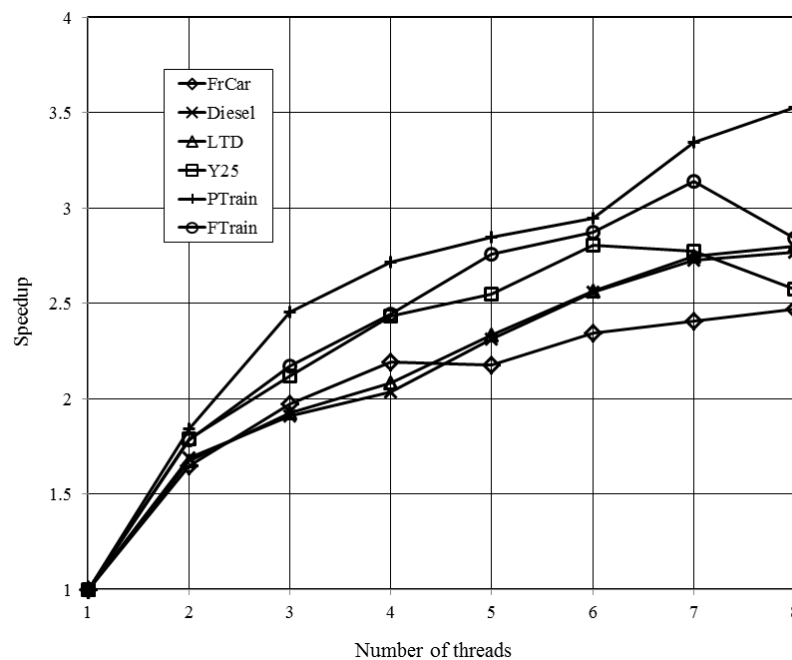
Fig. 7. Simulation speedup vs. number of parallel threads

The new rail profile R65 is used in the tangent track as well as for the inner rail in the curves. The worn rail profile R65 is assigned to the outer rail in the curves. The track irregularities for all simulations correspond to the track of a good quality according to the UIC standard.

The speedup effect of the modeling process versus the number of parallel threads is shown in Fig. 9. These results are obtained on a computer with an AMD Ryzen 7 2700 8-Core CPU (Processor 2).

4. EXAMPLE OF MULTI-VARIANT PARALLEL COMPUTATIONS

As discussed in the first part of this paper [1], there are several possible strategies to compute a series of numerical experiments: (1) to run one solver sequentially with allowed maximum of

computing threads in it, (2) to run allowed maximum of solvers with the only computing thread in it, or (3) to run several solvers with several threads in each. In UM software, the maximum number of simultaneously running processes (solvers) and threads in one solver is limited by the number of logical processors of the operating system. Let us analyze which computing strategy is the fastest one.
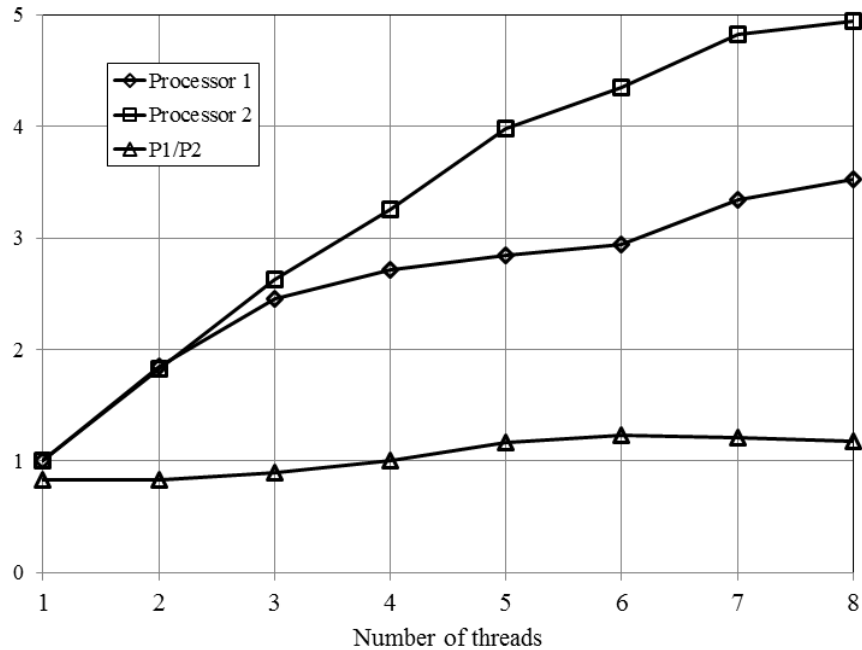

Fig. 8. Passenger train simulation speedup for Processors 1, 2 and T_h ratio (P1/P2) vs. number of parallel threads
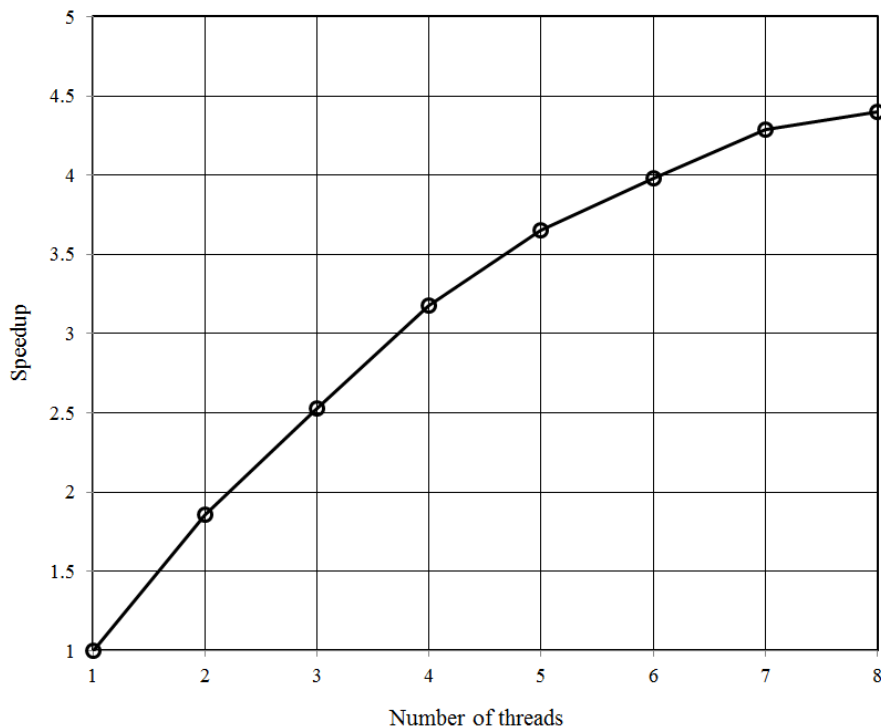

Fig. 9. Speedup of simulation of the wear of wheel profiles vs. number of parallel threads

Computing tests were done on the computer with Intel Core i7-4790 CPU, 4 GHz, which has 4 processor cores, and 8 GB RAM (Processor 1). For each processor core that is physically presented,

the operating system addresses two logical (virtual) cores. Architecturally, a processor with Hyper-Threading Technology by Intel consists of two logical processors per core, each of which has its own processor architectural state. Therefore, the processor has 8 logical processors.

For the computing tests, a model of a freight car with a three-piece bogie was used, Fig. 1. This model was taken for these tests as a typical model of a railway vehicle. The model is not too simple and not too complex, which makes it suitable for estimation of the effectiveness of computational strategies. A series of numerical experiments included 51 experiments of running in a 500-meter tangent track with different speeds starting from 30 to 80 km/h. Every numerical experiment takes from 30 to 90 seconds of CPU time. This CPU time is long enough to level the CPU time that is required for running process themselves and fluctuations in Windows background process activities. Comparison of the required CPU time for different strategies is reported in Table 3.

First of all, as shown in Tab. 3, the fastest strategy to finish the series of numerical experiments is using the maximum number of processes with the only thread in every process. The maximum speedup factor using 4 physical cores and 8 logical processors is 3.6.

Second, we can see that the Hyper-Threading Technology that divided 4 physical cores into 8 logical processors does not have a huge effect in these tests. Indeed, the best test case that utilized 8 processes x 1 thread ("8x1 configuration" for brevity) showed just nearly 7% better simulation time than the second fast 4x1 configuration.

Third, 4x8 and 8x8 configurations showed impractically high simulation time more than 10 times higher than the 1x1configuration. So we can see critical extra CPU expenses for managing processes and threads for the 4x8 and 8x8 configurations.

Table 3

Processes vs. threads.

Total time in seconds spent for the simulation of a series of 51 numerical experiments (speedup factor)

Processes	Threads		
	1	4	8
1	2735 (1)	1495 (1.8)	1745 (1.6)
4	813 (3.4)	1064 (2.7)	extremely slow
8	755 (3.6)	2980 (0.9)	extremely slow

5. CONCLUSION

Use of multicore processors for the simulation of dynamics of rail vehicles and trains speeds up the process 2.5-3.5 times in the case of a 4-core processor and up to 5 times for an 8-core one. The maximal effect is achieved for large models of trains running over flexible bridges.

Evaluation of wheel profile wear on a computer with an 8-core processor can be executed about 4.5 times faster than in the case of one-thread computations.

The most efficient method for speeding up multi-variant computations consists of parallelization of simulation processes in one-thread mode each. The maximum speedup factor using 4 physical cores and 8 logical processors is 3.6.

Acknowledgments

References

1. Pogorelov, D. & Rodikov, A. & Kovelev, R. Parallel computations and co-simulation in Universal Mechanism Software. Part I: Algorithms and implementation. *Transport Problems.* 2019. Vol. 14(3). P. 163-175.

2. *Universal Mechanism.* Bryansk: Laboratory of Computational Mechanics. Available at: www.universalmechanism.com/

3. Pogorelov, D.Y. Simulation of rail vehicle dynamics with Universal Mechanism software. In: Sładkowski, A. (ed.). *Rail vehicle dynamics and associated problems*. Gliwice: Wydawnictwo Politechniki Śląskiej. 2005. P. 13–58.

4. Nakajima, T. & Feldmeier, C. & Sugiyama, H. Flexible Moving track model for curve negotiation analysis of railroad vehicles with rail roll deflection. *IMechE Part K: Journal of Multi-body Dynamics*. 2016. Vol. 230. No. 1. P. 85-98.

5. Kovalev, R. & Lysikov, N. & Mikheev, G. & et al. Freight car models and their computer-aided dynamic analysis. *Multibody System Dynamics*. 2009. Vol. 22. No. 4. P. 399-423.

6. Wu, Q. & Spiryagin, M. & Cole, C. & et al. International benchmarking of longitudinal train dynamics simulators: results. *Vehicle System Dynamics*. 2018. Vol. 56. No. 3. P. 343-365.

7. Михеев, Г.В. & Погорелов Д.Ю. & Родиков, А.Н. Методы моделирования динамики железнодорожных колесных пар с учетом упругости. *Вестник Брянского государственного технического университета*. 2019. No. 4(77). P. 40–51. [In Russian: Mikheev, G. & Pogorelov, D. & Rodikov A. Methods for wheelset dynamic modeling taking into account elasticity. *Proceedings of Bryansk State Technical University*].