A parallel algorithm for the simulation of controlled motion of space tether systems

V V Zhidchenko\(^1\) and A N Kovartsev\(^1\)

\(^1\)Samara National Research University, Moskovskoe Shosse 34A, Samara, Russia 443086

e-mail: vzhidchenko@yandex.ru

**Abstract.** Simulation of the controlled motion of space tether systems differs significantly from the free motion case. Tether deployment increases its length and adds new points into the discrete model of the tether. A set of dynamic equations is expanded with equations that describe the motion of the new points. The necessary condition at each step is the consistency of the system. Tether deployment is supervised by the control system which limits the tether length. The problem arises of solving a system of differential equations with variable and increasing number of equations. In this paper a parallel algorithm for simulating the controlled motion of space tether systems is designed. The parallel algorithm provides an ability to split the tether to such number of points that the model becomes similar to the model with distributed parameters. As a result, the simulation of several modes of tether motion unavailable for other methods becomes possible. The estimation of speedup provided by the algorithm is implemented. It shows that the speedup is comparable with the more simple case of the free motion simulation. Computational experiments have been carried out for the algorithm performance evaluation.

1. Introduction

The study of the space tether systems (STS) dynamics is of great practical importance due to the wide range of possible applications of STS: the study of the upper atmosphere, the generation of artificial gravity on a spacecraft, transport operations in space and deorbiting the cargoes, changing the parameters of orbits of a spacecraft, electric power generation, probing the geomagnetic field of the Earth, space debris removal, etc. [1, 2, 3]

There are two types of motion of the STS. With the free motion, the two bodies are connected by a fixed-length tether. The task of controlled motion considers the change of the tether length in accordance with a given control function (control law). Modeling a controlled motion is obviously more difficult, since the control law affects the length and tension force of the tether. There are various models of controlled motion of the STS. In the model of an elastic rod, the tether is considered as a straight, inflexible, but tensile rod [4]. In the hinged-rod model, a tether is presented in the form of rigid rods connected by the hinge joints. The flexibility of the tether is considered, and also the influence of aerodynamic forces during the motion of the tether in the atmosphere is taken into account [5, 6]. In the bead model, the tether is represented by a set of material points with a finite mass, connected by the springs of zero mass [2, 7, 8, 9]. The bead model is simpler than the hinged-rod model. However, in order to achieve high accuracy, a large number of points are needed, which increases the complexity of computations. For this reason, most of the works that use the bead model deal with a few dozen points. On the other hand, with a large number of points, the bead model
outperforms other models in such aspects as accurate modeling of the deployment, consideration of uneven tension force or electrical conductivity of the tether. This paper aims to eliminate the limitations on the number of points when modeling the controlled motion of the STS using the bead model.

Partial differential equations with complex boundary conditions are used when modeling extended tether systems characterized by an uneven distribution of the mass characteristics of the system and environmental parameters in space. The complexity of the boundary conditions is caused by the presence of end bodies that perform spatial oscillations and by the variability of the tether length. The number of equations in the system of ordinary differential equations (ODE) describing the dynamic model of the STS can be measured in tens of thousands. This leads to a high computational cost of simulating the motion of the tether system on a computer. The numerical solution of such systems is difficult, even for modern computing systems.

This paper presents the results of the development of a parallel algorithm for the numerical integration of the controlled motion of space tether systems. This algorithm uses the method of parallelization of the fourth-order Runge-Kutta method, similar to the one proposed in [10]. However, the algorithm under consideration differs significantly from the algorithm that simulates the free motion of the tether. Initially the spacecraft and the satellite are interconnected by a very small fragment of the tether. As the tether deploys, new points are added, and the corresponding equations describing the motion of new tether points are added to the system. A problem arises of adding new points in such a way that the basic laws of mechanics are preserved. In controlled motion, the tension-control law implemented by the deployment system must also be taken into account. All this leads to the problem of integrating an ODE with a variable and increasing number of equations. The concepts of the parallel algorithm for implementing the controlled motion of the STS have changed significantly. A new model of the parallel algorithm was created for integrating the controlled and free motion of the tether, which acceleration is comparable to the case of the algorithm for the free motion.

2. Mathematical model of controlled motion of the space tether system

Consider the task of the tether deployment. When constructing a mathematical model, the STS is considered as a mechanical system consisting of a set of $n$ material points interconnected by elastic weightless links (Figure 1).

![Figure 1. Tether deployment scheme.](image)

The deployment of the tether is performed from the spacecraft. Control under the tether deployment is implemented by the control system (CS) in accordance with the deployment program using a tether braking mechanism. In the beginning of the deployment process, there are two material points: the spacecraft and the satellite. After the tether has been deployed to a predetermined length of the tether segment, a new segment of the tether is fixed, i.e. a new material point appears. Then, the next cycle of tether deployment is performed on the short segment of the tether adjacent to the spacecraft. The process is repeated until the tether deploys to the specified length.

2.1. Mathematical model of the motion of internal points of the STS

The mathematical model of the motion of internal points of the STS is described by the following system of ODEs:
\[
\begin{align*}
\frac{dR_i}{dt} &= V_i \\
\frac{dV_i}{dt} &= -\frac{\mu_i}{r_i^3} R_i - \frac{F_i}{m_i L_i} (R_i - R_{i-1}) + \frac{F_{i+1}}{m_i L_i} (R_{i+1} - R_i)
\end{align*}
\]
\[
R_i = [x_i, y_i, z_i]^T, V_i = [V_{ix}, V_{iy}, V_{iz}]^T, r_i = |R_i|, L_i = |R_{i+1} - R_i|,
\]
\[
F_i = \begin{cases} c \frac{L_{i-1} - l_{i-1}}{l_{i-1}}, & L_{i-1} > l_{i-1} \\
0, & L_{i-1} \leq l_{i-1}
\end{cases}
\]

where \( F_i \) is the tension force of the tether, acting on the \( i \)-th inner point and determined by Hooke's law, \( \mu \) is the gravitational constant of the Earth, \( R_i \) and \( V_i \) are the coordinates and the velocity of the \( i \)-th point in the geocentric fixed coordinate system.

### 2.2. Mathematical model of the satellite motion

The satellite interacts with the spacecraft at the first stage of the tether deployment only, until the moment when the tether length has reached the limit of a new point formation. The mathematical model of the satellite motion is described by the following system of ODEs:

\[
\begin{align*}
\frac{dR_0}{dt} &= V_0 \\
\frac{dV_0}{dt} &= -\frac{\mu_0}{r_0^3} R_0 - \frac{F_0}{m_0 L_0} (R_0 - R_0) \\
r_0 &= |R_0|, L_0 = |R_1 - R_0|
\end{align*}
\]

### 2.3. Mathematical model of the spacecraft motion

The mathematical model of the spacecraft motion at the deployment stage is described by the following system of ODEs:

\[
\begin{align*}
\frac{dR_n}{dt} &= V_n \\
\frac{dV_n}{dt} &= -\frac{\mu_n}{r_n^3} R_n + \frac{F_n}{m_n L_n} (R_{n-1} - R_n) \\
r_n &= |R_n|, L_n = |R_{n-1} - R_n|, F_n = c \frac{L_n - l_n}{l_n}
\end{align*}
\]

The magnitude of the unstretched tether length (deployed) \( l_n \) is determined separately in accordance with the deployed section defined by the CS and described by a separate system of ODEs describing the operation of the deployer mechanism.

### 2.4. Model of the deployer mechanism

The process of tether deployment is described by a one-dimensional model described by the following system of ODEs:
\[
\begin{align*}
\frac{dl}{dt} &= V_l \\
\frac{dV_l}{dt} &= \frac{1}{m_u} (F_n - F_c) \\
F_c &= p_l \left( \sum_{i=0}^{n} l_i - l_p \right) + p_V (V_l - V_{lp})
\end{align*}
\]  

(4)

Here \( l \) is the length of the first section of the tether, counting from the spacecraft (metric, unstretched), \( V_l \) is the release rate of the tether, \( m_u \) is the coefficient that takes into account the inertia of the deployer mechanism, \( F_c \) is the force acting on the tether in the deployer mechanism, \( F_n \) is the tension force of the tether acting on the spacecraft, \( l_p \) and \( V_{lp} \) are the predefined (nominal) functions of tether length and deployment rate depending on time that define the braking force, \( p_l \) and \( p_V \) are the feedback factors; \( l \) is the unstretched length of each section.

The nominal law for controlling the deployment is defined by the following system:

\[
\begin{align*}
\frac{d\theta}{dt} &= \omega_\theta \\
\frac{d\omega_\theta}{dt} &= -2 \frac{V_{lp}}{l_p} (\omega_\theta + \Omega) - \frac{3}{2} \Omega^2 \sin(2\theta) \\
\frac{dl_p}{dt} &= V_{lp} \\
\frac{dV_{lp}}{dt} &= l_p \left( (\omega_\theta + \Omega)^2 - \Omega^2 (1 - 3 \cos^2 \theta) \right) - \frac{T_0 (l_p, V_{lp})}{m_0}
\end{align*}
\]  

(5)

where \( \Omega \) is the angular velocity of the orbital motion of the spacecraft in a circular orbit, \( m_0 \) is the satellite mass; \( \theta \) is the angle of deviation of the tether from the direction of the local geo-vertical; \( l_p \) is the nominal length and \( V_{lp} \) is the deployment rate of the tether, \( T_0 \) is the tension force of the tether (nominal); \( \omega_\theta \) is the angular velocity of the tether rotation over the spacecraft.

The nominal (controlled) tension force is given by the corresponding dynamic law:

\[
T_0 = f (l_p, V_{lp})
\]  

(6)

2.5. Algorithm for the new point formation

The process of the tether deployment is followed by the sequential discretization of the motion simulation problem. It is assumed that finally the tether system should consist of \( N \) points. The discretization step determines the accuracy of the calculations and is selected in advance. The distance between the points is \( l_0 = L_T / (N - 1) \), where \( L_T \) is the length of the tether. Initially, the spacecraft and the satellite are interconnected by a small (tending to zero) section of the tether. As the tether is being deployed and it is approaching approximately the specified length of the section to be formed \( (l_0) \), it becomes necessary to generate a new point. The introduction of the new point should be done in such a way as not to violate the law of conservation of energy of the mechanical system before and after adding a new point. Figure 2 shows the scheme for adding a new point.
Figure 2. Scheme for adding a new tether point.

In figure 2, $\vec{r}_0$ is the radius vector and $\vec{v}_0$ is the velocity of the spacecraft; $\vec{r}_1$ is the radius vector and $\vec{v}_1$ is the velocity of the existing tether point in the geocentric coordinate system. The position of this point relative to the spacecraft and its relative velocity are as follows:

$$\begin{align*}
\Delta \vec{r}_1 &= \vec{r}_1 - \vec{r}_0 \\
\Delta \vec{v}_1 &= \vec{v}_1 - \vec{v}_0
\end{align*}$$

The position of the new point relative to the spacecraft is determined as follows:

$$\Delta \vec{r} = \frac{|\Delta \vec{r}_1| - l_0}{|\Delta \vec{r}_1|} \Delta \vec{r}_1, \quad \vec{r} = \vec{r}_0 + \Delta \vec{r}$$

Let $\vec{n}$ and $\vec{t}$ be the unit normal and tangential vectors accordingly, along the length of the tether, locally in the vicinity of the spacecraft. Denote by $\alpha$ the angle between the vectors $\Delta \vec{r}_1$ and $\Delta \vec{v}_1$. In order to find the component of the relative velocity of the new point, directed along the tether, we decompose the velocity $\Delta \vec{v}_1$ into tangential and normal components:

$$\Delta \vec{v}_1 = \Delta \vec{v}_{1t} + \Delta \vec{v}_{1n}$$

The tangential and normal components can be calculated from the following equations:

$$\Delta \vec{v}_{1t} = |\Delta \vec{v}_1| \frac{\Delta \vec{r}}{|\Delta \vec{r}|} \cos(\alpha), \quad \cos(\alpha) = \frac{(\Delta \vec{r}_1, \Delta \vec{v}_1)}{|\Delta \vec{r}_1| |\Delta \vec{v}_1|}, \quad \Delta \vec{v}_{1n} = \Delta \vec{v}_1 - \Delta \vec{v}_{1t}$$

Taking into account the equality of the tangential component of the velocity at the new point and at the last one on the tether, it is possible to determine the velocity vector at the new point:

$$\Delta \vec{v}_n = \frac{|\Delta \vec{r}|}{|\Delta \vec{r}_1|} \Delta \vec{v}_{1t}, \quad \Delta \vec{v} = \Delta \vec{v}_n + \Delta \vec{v}_{1t}, \quad \vec{v} = \vec{v}_0 + \Delta \vec{v}$$

The obtained result should be corrected with the assumption that the spacecraft mass has decreased by the amount of the mass of the new segment of the tether $m_i$: 
The mass of the spacecraft should be reduced by the value of \( m_i \). The length of the new segment can be calculated by the formula:

\[
\Delta l_i = l \frac{|\Delta v_i|}{|\Delta r_i|}
\]

The rest part of the tether adjacent to the spacecraft has the following length:

\[
l_{\infty} = l \frac{|\Delta r|}{|\Delta r_i|}
\]

The condition for the formation of a new point can be defined as follows:

\[l > l_0 + 0.25 \cdot l_0\]

3. Model of the parallel algorithm for the simulation of controlled motion of the STS

3.1. Description of the tether segmentation scheme for the case of controlled motion of the STS

A feature of the considered case of controlled motion of the STS is the uneven distribution of the computational complexity as the tether is deployed. At the initial moment of the calculations, there are two moving objects: the spacecraft and the satellite. Obviously, the task of their motion simulation does not require significant computational resources. However, as the tether is deployed, when the number of tether points increases to several hundred or thousands, the complexity of the computational problem also increases.

The concept of tether segmentation, proposed at the first stage of research and described in [10], has been changed to take into account the growth of the complexity of the problem being solved. In the current work, the points are numbered from the satellite to the spacecraft. The reason is that new points are added to the tether at certain intervals along its deployment. In order to prevent copying the elements of the matrix that contains the positions of the objects of the tether system (the points, the satellite and the spacecraft) when adding a new point, the position of the new point is written in the place of the spacecraft. The position vector of the spacecraft is copied into the next line.

Tether deployment is simulated in the "zero" parallel process, which, apart from the simulation of the STS motion, organizes the parallel calculations. It performs the functions of a "manager" of the computational process. As the matrix of phase coordinates of the STS is filled in process 0, this process copies the new part of the tether points into a free process and launches calculations on it. A single tether point and the spacecraft remain in the process 0.

Figure 3 shows the tether segmentation scheme for the case of controlled motion of the STS. The proposed parallel algorithm of controlled motion of the tether starts as a sequential program in a single process. At the moment of filling the matrix of phase coordinates in process 0, almost the entire matrix, except the last two points, is transferred to the process 1. After this, the tether deployment algorithm is still implemented in process 0. In process 1 the free motion of the tether is computed [10]. At the last stage of tether deployment, all processes are launched, and upon completion of tether deployment in process 0, the proposed parallel computational scheme simulates the free motion of the STS.

3.2. Visual model of the parallel algorithm

We use the notation of the graph-symbolic programming (GSP) technology [11] in order to visually represent the proposed parallel algorithm. GSP technology allows describing the algorithm as a set of
control flow diagrams, and then automatically compiling and running a program that implements this algorithm. The control flow diagram is a directed graph in which the nodes indicate the actions performed on the data of a certain subject area, and the arcs indicate the sequence of these actions.

![Figure 3](image_url)  
Figure 3. Tether segmentation scheme for the case of controlled motion.

If several arcs emanate from a node, then depending on the type of the arcs, either one or several adjacent nodes into which these arcs lead can be executed. Arcs of different types are graphically designated in different ways. The advantage of control flow diagrams is the visual representation of the sequence of calculations in the program. The disadvantage is the lack of displaying data dependencies between the nodes. Figure 4 shows a fragment of the parallel algorithm designed to solve the system of differential equations that describe the controlled motion of STS using the fourth-order Runge-Kutta method. The coefficients for the fourth order Runge-Kutta method are calculated in four phases (four stages).

The arcs marked with a circle indicate the transfer of control to a new parallel process. The arcs marked with a slash indicate the return of control from a parallel process. In each phase of the calculations, four processes are used to calculate the different tether segments.

The first tether segment adjacent to the spacecraft is calculated in deployment modules (nodes 1.1, 2.1, 3.1, 4.1, 1.1.0, 2.1.0, 3.1.0, 4.1.0, 5.1.0, 6.1.0, in figure 4) using the mathematical model for calculating the controlled motion of the tether described above. In addition, these modules manage the process of starting calculations on other tether segments. The processes described by the nodes 1.2–4.2, 1.3–4.3, 1.4–4.4, are started as the required number of points accumulates on the deployment segment. In this case, the accumulated array of points is transferred to the free process and deleted from the deployment process. The deployment module, represented by the nodes 1.1, 2.1, 3.1, 4.1, calculates the Runge-Kutta coefficients for the corresponding phase of the algorithm. In module 4.1, in addition to the motion simulation of the STS, the tether deployment mechanism is simulated and the program for adding new tether points is executed. The deployment module also monitors the distribution of data across the processes and is responsible for starting other processes. It plays the role of a manager, determining the load of the processes of a parallel program. At the end of each time step calculation, a message is generated in the deployment module for each of other processes, which represents the number of tether segments formed. This value is stored in the variable Fupr. Messages are sent from node 5.1.0 to nodes 5.X.0 (X∈ {2,3,4}). Depending on the content of the received message, each process either continues to sleep, or starts execution and receives its tether segment.
from the deployment module, or changes the numbers of its communication neighbors in accordance with the diagram presented in figure 3. When the last simulation time step is reached the deployment module generates a signal to stop the calculations. The remaining modules in the processes 2, 3, 4 operate in accordance with the models describing the free motion of the STS for the corresponding segments.

In figure 4 some nodes are marked with icons with a simplified image of the space tether system and the number of the computation phase. These vertices are sequential subroutines for calculating the coefficients of the Runge-Kutta method for individual tether segments. Thus, the graph of the algorithm in figure 4 is hierarchical. The hierarchical composition of graphs of algorithms in GSP technology allows hiding irrelevant details of the implementation of various parts of the algorithm, leaving only important elements on the graph in terms of understanding the structure of the algorithm.

The dashed arcs in figure 4 show the messages sent between the processes. Messages send data and describe synchronization between the processes. The operation of receiving a message is blocking, that is, the receiving process is suspended until the message, represented by an arc in the graph, is received. The modules X.Y.0 (X∈\{1,2,3,4\}, Y∈\{1,2,3,4\}) at each phase of the calculations perform the exchange of boundary points between the processes that simulate the motion of the adjacent tether segments. The process numbers of the recipients of messages vary depending on the quantity of formed tether segments. In figure 4, messages corresponding to each value of the quantity of tether segments are indicated by arcs of a unique type.

**Figure 4.** Parallel algorithm for calculating the coefficients of the Runge-Kutta method.
3.3. Estimation of the acceleration of the parallel algorithm for numerical simulation of controlled motion of the STS

This section presents the estimation of the acceleration of a parallel program that implements the proposed algorithm. The actions described by the modes of the same phase of calculations, which are performed simultaneously in different parallel processes and denoted by the numbers X.1, X.2, X.3, X.4 in figure 4 (where X ∈ {1,2,3,4} is the phase number), process tether segments of the same length. The duration of these calculations is approximately the same in all processes within the same phase. The nodes of process 0, which simulate the operation of the control system and the motion of tether points directly adjacent to the spacecraft, process a variable number of points (from 2 to K). At the stage of tether deployment, the average number of processed points in process 0 is less than in other processes, therefore process 0 runs faster than other processes in each phase. As the number of points in the deploying first tether segment approaches the number of points in other segments, the duration of the calculations in all processes is equalized. Before the separation of the next section of the tether, the process 0 runs longer than the others, since it simulates the process of deployment and the motion of a large number of points which is close to the number of points in other processes. Nevertheless, the trace of the program shows that these fluctuations have little effect on the average duration of the execution of the nodes in the process 0. Denote this value by $T_0$. The duration of the execution of the nodes in other processes varies less and amounts to $T_i \approx 1.132T_0$.

The simulation of the controlled motion of the STS in the considered formulation of the problem includes three stages:

1) the initial deployment stage, in which the first tether segment extends, and the calculations are performed sequentially in process 0;
2) the deployment stage, at which the modeling of the control system continues and the generation of new tether points is carried out with the transmission of the tether segments to other processes;
3) the stage of free motion in which the deployment is stopped and the free motion of the full length of the tether is simulated in parallel in all processes.

At each stage, the values of the phase coordinates for the boundary points of the tether segments processed in the respective processes are transmitted between the processes. The length of these messages is fixed, so the duration of the message exchange between two processes at each stage does not depend on the stage number and the length of the tether segments in the processes. If we assume that the messages between the processes that form the communication pairs can be transmitted simultaneously, then the total duration of the messages exchange at each stage does not depend on the number of processes. The trace of the program confirms this assumption.

The duration of the sequential program can be estimated by the following expression:

$$T_s = 4T_0N_0^d + (4T_0 + 4T_i)N_1^d + (4T_0 + 2(4T_i))N_2^d + \ldots + (4T_0 + (p-1)4T_i)N_{p-1}^d + 4T_i pN_f$$  \hspace{1cm} (7)

where $N_0^d$ is the number of simulated time steps during which the first tether segment is deployed, $N_i^d$ is the number of time steps during which the $i$-th tether segment is deployed, $N_f$ is the number of time steps during which the free motion of the tether is simulated, $p$ is the number of tether segments, equal to the number of processes in the parallel program. In the case of uniform deployment of the tether, using the formula of arithmetic progression, the expression (7) can be reduced to the following form:

$$T_s = 4T_0N_0^d + 4T_0 \sum_{i=1}^{p} N_i^d + 2p(p-1)T_i N_i^d + 4T_i pN_f = 4T_0N_0^d + 4T_i pN_f$$ \hspace{1cm} (8)

where $N_0^d$ is the total number of time steps during which the tether is deployed.

The estimate of the duration of the parallel program can be represented by the following expression:

$$T_p = (4T_0 + \beta K)N_0^d + (p - 1) \frac{\beta K}{2} + (4T_i + 5\beta)(N_f - N_0^d)$$ \hspace{1cm} (9)

where $N_0^d$ is the total number of time steps during which the tether is deployed.
where $\beta$ is the duration of the exchange of messages at each stage, $K$ is the number of points in each tether segment, $N^T$ is the total number of time steps during which the motion of the STS is simulated.

The second term in expression (9) estimates the duration of the transmission of the tether segment to the process responsible for its simulation. In total, there are $(p-1)$ such transmissions. The duration of each transmission does not take into account the length of the communication protocol packets. It is estimated as the sum of the transmission durations for each point. The value of $\beta$ is taken as the duration of the transmission of two messages each containing a single point, since the processes at each phase exchange their boundary points in pairs.

Acceleration and efficiency of the parallel program are estimated by the following expressions:

$$S = \frac{T_s}{T_b} = \frac{4T_0 N^d + 2p T_i (N^d - N_0^d) + 4 T_i p N^f}{(4T_0 + \beta)N_0^d + (p-1)\frac{B K}{2} + (4T_i + 5\beta)(N^T - N_0^d)}$$  \hspace{1cm} (10)$$

$$E = \frac{S}{p} = \frac{4T_0 N^d p^{-1} + 2p T_i (N^d - N_0^d) + 4 T_i N^f}{(4T_0 + \beta)N_0^d + (p-1)\frac{B K}{2} + (4T_i + 5\beta)(N^T - N_0^d)}$$  \hspace{1cm} (11)$$

With the increase in the number of points in each tether segment, the acceleration will be defined by the ratio of the time during which the tether is deployed to the total period of time of STS motion being simulated:

$$S \rightarrow \frac{4N^d + 2p(N^d - N_0^d) + 4 p N^f}{4N_0^d + 4(N^T - N_0^d)}$$  \hspace{1cm} (12)$$

The more part of the time interval is occupied by the free motion of the tether, the closer the acceleration is to the number of processes of the parallel program.

4. Computational experiments

Using the graph-symbolic programming technology, the proposed parallel algorithm was implemented as a program for a computing cluster. MPI technology was used for the message passing between the parallel processes. The program was compiled and run on the high-performance computing cluster "Sergey Korolev" of the Samara University. The simulation of the controlled motion of the STS was carried out for the case with the ratio of deployment time to the total simulated time interval equal to 0.425. Figure 5 shows the dependences of the acceleration and efficiency of the parallel program consisting of four processes on the number of points in the tether segment simulated on each process.

![Figure 5. Dependence of acceleration (S) and efficiency (E) of the parallel program on the number of points in each tether segment.](image)
Lines "S" and "E" correspond to the estimates calculated by formulas (10) and (11); the points "Sreal" and "Ereal" correspond to the experimental data. Due to the overestimation of the duration of message transmission mentioned above, the value of the theoretical estimate of the acceleration is lower than the real value at some points. In general, the computational experiment has showed that the proposed estimates of acceleration and efficiency correspond to real measurements.

5. Conclusion
The paper presents a parallel algorithm for modeling the controlled motion of the STS, which allows using an arbitrarily large number of points in the bead model. An increase in the number of points makes it possible to bring the model under consideration to a model of a system with distributed parameters and to perform the simulation of various processes occurring in the STS, with accuracy unattainable for other methods. The estimates of the acceleration and efficiency of parallel programs implementing the described algorithm are given. The agreement of the theoretical estimates with the actually observed dependencies has been verified experimentally. The principle of organizing parallel processing when solving a system with a variable number of differential equations proposed in the work can be applied not only in the bead model, but also in other models. This makes it possible to use the advantages of modern computing systems with parallel architecture to improve the accuracy of the simulation of controlled motion of the STS.

6. References
[1] Beletsky V V and Levin E M 1993 *Dynamics of space tether systems* (San Diego: Univelt, Inc.)
[2] Cosmo M L and Lorenzini E C 1997 *Tethers in Space Handbook* (Smithsonian Astrophysical Observatory)
[3] Johnson L, Gilchrist B, Estes R D and Lorenzini E C 1999 Overview of future NASA tether applications *Adv. in Space Res.* **24**(8) 1055-1063
[4] Williams P 2006 In-plane payload capture with an elastic tether *J. of Guidance, Control, and Dynamics* **29**(4) 810-821
[5] Williams P 2008 Deployment/retrieval optimization for flexible tethered satellite systems *Nonlinear Dynamics* **52**(1-2) 159-179
[6] Zhong R and Zhu Z H 2011 Dynamic analysis of deployment and retrieval of tethered satellites using a hybrid hinged-rod tether model *Int. J. of Aerospace and Lightweight Structures* **1**(2) 239-259
[7] Zabolotnov Y M 2015 Control of the deployment of a tethered orbital system with a small load into a vertical position *J. of Appl. Mathematics and Mechanics* **79**(1) 28-34
[8] Wang C and Zabolotnov Y M 2017 Management at deploying a tethered system in elliptical orbit *Izvestiya Samarskogo nauchnogo centra RAN* **19**(1) 91-97
[9] Dong Z, Zabolotnov Y M and Wang C 2017 Modeling and analysis of deployment dynamics of a distributed orbital tether system with an atmospheric sounder *Adv. in the Astronautical Sciences* **161** 1053-1063
[10] Kovartsev A N and Zhidchenko V V 2017 Application of Graph Models to the Parallel Algorithms Design for the Motion Simulation of Tethered Satellite Systems *Proc. Int.Conf. on Parallel Comp. Technol. LNCS* **10421** 114-123
[11] Egorova D and Zhidchenko V 2015 Visual Parallel Programming as PaaS cloud service with Graph-Symbolic Programming Technology *Proc. Inst. Syst. Program* **27**(3) 47-56

Acknowledgements
The work was partially funded by the Russian Federation Ministry of Education and Science and Russian Foundation of Basic Research. Grant #16-41-630637.