

Использование графовых моделей при разработке параллельных алгоритмов моделирования движения космических тросовых систем

А.Н. Коварцев^а, В.В. Жидченко^а

^а Самарский национальный исследовательский университет имени академика С.П. Королёва, 443086, Московское шоссе, 34, Самара, Россия

Аннотация

Протяженные космические тросовые системы (КТС) характеризуются неравномерным распределением массовых характеристик системы и параметров окружающей среды в пространстве, что вызывает необходимость применения математических моделей с распределенными параметрами. Для моделирования используются дифференциальные уравнения в частных производных со сложными краевыми условиями. Сложность краевых условий вызвана наличием конечных тел, совершающих пространственные колебания, и переменностью длин тросов. Это приводит к большим временным затратам при математическом моделировании движения КТС на ЭВМ.

В работе приводится параллельный алгоритм моделирования движения КТС и его реализация в виде графовой модели в технологии графо-символического программирования. Рассматриваются основные характеристики предложенного алгоритма, а также преимущества использования графических моделей алгоритмов при моделировании движения КТС.

Ключевые слова: космическая тросовая система; параллельные вычисления; графические модели; визуальное программирование

1. Введение

Применение космических тросовых систем (КТС) открывает новые возможности в использовании космического пространства: создание искусственной тяжести, транспортные операции в космосе, возвращение полезных грузов с орбиты, запуск малых спутников с базового космического аппарата (КА), использование геомагнитного поля Земли для орбитальных маневров, создание орбитальных электростанций, зондирование атмосферы, геомагнитного и гравитационного полей, удаление космического мусора и т.д [1].

В настоящее время, несмотря на наличие большого количества работ, в которых рассматриваются различные аспекты проектирования космических тросовых систем, существует определенный дефицит научных исследований, посвященных разработке методов анализа и синтеза управляемого и свободного движения космических тросовых систем большой протяженности. Протяженные тросовые системы характеризуются неравномерным распределением массовых характеристик системы и параметров окружающей среды в пространстве, что вызывает необходимость применения математических моделей с распределенными параметрами. В данном случае требуется использование аппарата дифференциальных уравнений в частных производных со сложными краевыми условиями. Сложность краевых условий вызвана наличием конечных тел, совершающих пространственные колебания, и переменностью длин тросов. Все это приводит к большим временным затратам при математическом моделировании движения тросовой системы на компьютере. В данном случае мы сталкиваемся с системами обыкновенных дифференциальных уравнений (ОДУ) больших размерностей. Число уравнений измеряется десятками тысяч. Численное решение таких систем напрямую представляется затруднительным даже для современных вычислительных ресурсов.

Существующие подходы к решению больших систем уравнений можно условно разделить на два основных класса. Первое направление сосредотачивается на распараллеливании известных численных методов (зачастую без изменения самих методов) [2]. Однако результаты решения систем ОДУ, получаемые при таком распараллеливании на кластерных системах, не являются впечатляющими, поскольку в них не учитываются какие-либо особенности решаемой задачи (например, слабая связанность подсистем).

Во втором классе подходов содержатся методы, сокращающие вычислительные затраты за счёт специальных эвристических приемов, обычно использующих физические особенности решаемой задачи. При этом точности решения уделяется меньше внимания. Увеличение скорости работы алгоритмов расчёта происходит за счёт снижения точности численных методов без понижения качества получаемых результатов.

Рассмотрим возможность применения указанных подходов к задаче моделирования движения космической тросовой системы.

2. Математическая модель движения тросовой системы

Общепризнанной и одной из наиболее полных математических моделей движения тросовой системы является модель, в которой трос описывается системой дифференциальных уравнений в частных производных. В этом случае для описания движения троса используются математические модели механики сплошной среды, в которых трос рассматривается как растяжимое (или нерастяжимое) тонкое тело, чаще всего, большой длины [1]. Вывод уравнений движения такой системы достаточно прост и заключается в выделении на тросе элементарного участка длиной ΔS и в применении к нему второго закона Ньютона. В конечном итоге несложно получить уравнение:

$$\rho(S) \frac{\partial^2 \vec{r}}{\partial t^2} = \frac{\partial \vec{T}}{\partial S} + \vec{q} \quad (1)$$

где $\rho(S)$ - линейная плотность троса, \vec{r} - радиус-вектор элементарного участка троса, t - время, \vec{T} - сила натяжения троса, \vec{q} - равнодействующая сила, действующая на элементарный участок, отнесенная к единице длины (распределенная нагрузка).

Для гибкого троса, который не воспринимает поперечных нагрузок, сила натяжения направлена по касательной к линии троса, поэтому

$$\vec{T} = T \vec{\tau}, \quad \vec{\tau} = \frac{1}{\gamma} \frac{\partial \vec{r}}{\partial S} \quad (2)$$

где $\vec{\tau}$ - единичный вектор касательной, $\gamma = \left| \frac{\partial \vec{r}}{\partial S} \right|$.

Закон растяжимости троса $T(\gamma)$ в самом простом случае представляет собой закон Гука

$$T(\gamma) = EA(\gamma - 1) \quad (3)$$

где E - модуль упругости Юнга, A - площадь поперечного сечения троса, $\gamma - 1$ - относительное удлинение троса.

Таким образом, уравнение движения гибкого троса (1) с учетом соотношений (2), (3), представляет собой уравнение в частных производных, которое решается при заданных граничных и начальных условиях движения.

Для интегрирования уравнения движения (1) могут быть использованы обычные численные методы решения дифференциальных уравнений в частных производных волнового типа: метод конечных разностей, метод разделения переменных и др. Эффективным представляется сведение решения уравнения в частных производных (1) к решению большого количества обыкновенных дифференциальных уравнений. В этом случае дискретизация системы (1) сначала производится по длине троса, а потом численно решается система обыкновенных дифференциальных уравнений для вычисления перемещений по времени узловых точек, на которые дискретно разбивается трос.

При интегрировании по времени исходной системы уравнений возникает довольно сложная в вычислительном отношении задача. Адекватное решение этой задачи требует тщательного выбора параметров используемых численных методов.

Дискретным аналогом уравнения (1) является механическая система, состоящая из совокупности N материальных точек, соединенных упругими невесомыми связями (рис. 1).

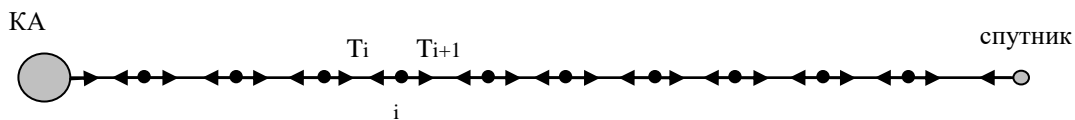


Рис. 1. Дискретная модель космической тросовой системы.

Различные модификации этой модели использовались многими авторами [3,4,5]. В рамках указанной модели учитываются основные эффекты движения КТС: растяжимость троса, распространение продольных и поперечных колебаний, диссипация энергии и др. При этом достаточно просто задаются граничные и начальные условия движения КТС через определение соответствующих условий для концевых точек троса. В качестве концевых тел могут быть рассмотрены как материальные точки, так и тела конечных размеров, для которых при необходимости записываются обычные уравнения их вращательного движения (динамические и кинематические уравнения Эйлера).

Уравнения движения совокупности M материальных точек с упругими связями записываются в виде системы обыкновенных дифференциальных уравнений

$$\frac{d\vec{r}_k}{dt} = \vec{V}_k, \quad m_k \frac{d\vec{V}_k}{dt} = \vec{F}_k \quad (4)$$

где \vec{r}_k , \vec{V}_k и m_k - радиус-вектор, скорость и масса k - ой материальной точки; \vec{F}_k - равнодействующая сил, действующих на материальные точки, $k = 1, 2, \dots, M$.

Для определения силы упругости троса в самом простом случае используется закон Гука, предполагающий линейную зависимость силы от растяжения. Если длина троса становится меньше его длины в недеформированном состоянии, то сила упругости принимается равной нулю, поскольку тонкий трос не работает на сжатие.

3. Численное моделирование движения космической тросовой системы

Рассмотрим некоторые особенности численного моделирования КТС как системы с распределенными параметрами. Для этого будем использовать многоточечную модель движения КТС (рис. 1). Представление троса как совокупности материальных точек сводит задачу интегрирования дифференциального уравнения в частных производных (1), описывающего движение КТС, к интегрированию системы обыкновенных дифференциальных уравнений высокого порядка. Данная задача численного интегрирования представляет собой классическую начальную задачу Коши

$$\frac{dX}{dt} = F(X, t), \quad (5)$$

где X - матрица фазовых координат для каждой точки КТС:

$$X = \begin{pmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \dots & \dots & \dots & \dots \\ x_{M,1} & x_{M,2} & \dots & x_{M,n} \end{pmatrix}. \quad (6)$$

Здесь n – количество переменных, описывающих состояние системы, M – суммарное количество точек троса, включая базовый аппарат и спутник.

$$F = \begin{pmatrix} f_{1,1}(X, t) & f_{1,2}(X, t) & \dots & f_{1,n}(X, t) \\ f_{2,1}(X, t) & f_{2,2}(X, t) & \dots & f_{2,n}(X, t) \\ \dots & \dots & \dots & \dots \\ f_{M,1}(X, t) & f_{M,2}(X, t) & \dots & f_{M,n}(X, t) \end{pmatrix} \quad (7)$$

Начальные условия описываются матрицей начальных условий $X(t_0) = X_0$, а моделирование движения КТС рассматривается на отрезке $t \in [t_0, t_K]$.

Обычно для численного решения уравнения (5) используют метод Рунге-Кутты четвертого порядка. При этом изменение фазовых координат производится по формуле:

$$X^{(r+1)} = X^{(r)} + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4), \quad (8)$$

где K_1, K_2, K_3, K_4 - матрицы коэффициентов формулы Рунге-Кутты.

4. Параллельный алгоритм численного моделирования движения тросовой системы

Очевидно, что для более точного решения системы уравнений (5) необходимо увеличить количество точек дискретизации троса. При $M \rightarrow \infty$ дискретный аналог троса превращается в непрерывную модель. Однако большое количество точек разбиения троса существенно увеличивает трудоемкость задачи моделирования его движения. Следовательно, необходимо произвести распараллеливание алгоритма решения задачи и использовать многопроцессорные ЭВМ. При этом возникает ряд трудностей. Во-первых, сказывается наличие связей каждой точки со своими правым и левым соседями, которые необходимы для учета сил натяжения троса (с точки зрения влияния гравитационных и инерциальных сил точки не зависят друг от друга). Во-вторых, матрицы K_1, K_2, K_3, K_4 не могут быть вычислены независимо друг от друга, а только последовательно в порядке увеличения индексов.

Пусть $X = X^{(r)}$ - начальная матрица фазовых координат. Для каждой j -ой точки троса на первом этапе (в первой фазе) матрица K_1 вычисляется по формулам:

$$k_1^j = f_j(x_{j-1}, x_j, x_{j+1}), \quad (9)$$

где $k_1^j, f_j(), x_{j-1}, x_j, x_{j+1}$ - строки соответствующих матриц K_1, F, X . Новое значение фазовых координат, необходимое для вычисления матрицы K_2 , можно представить в матричной форме

$$\tilde{X} = X + hK_1 / 2 \quad (10)$$

По аналогии с (9) и (10) формулы Рунге-Кутты для фаз 2, 3, 4 могут быть представлены следующими формулами:

$$k_2^j = f_j(\tilde{x}_{j-1}, \tilde{x}_j, \tilde{x}_{j+1}),$$

$$\tilde{\tilde{X}} = X + hK_2 / 2,$$

для второй фазы.

$$k_3^j = f_j(\tilde{\tilde{x}}_{j-1}, \tilde{\tilde{x}}_j, \tilde{\tilde{x}}_{j+1}),$$

$$\tilde{\tilde{\tilde{X}}} = X + hK_3,$$

для третьей фазы.

$$k_4^j = f_j(\tilde{\tilde{\tilde{x}}}_{j-1}, \tilde{\tilde{\tilde{x}}}_j, \tilde{\tilde{\tilde{x}}}_{j+1}),$$

в совокупности с (8) для четвертой фазы.

Разобьем трос на p сегментов. Положим, что M точек троса нацело делится на p так, что $m = M / p$. Таким образом, каждый сегмент троса содержит m точек (рис. 2).

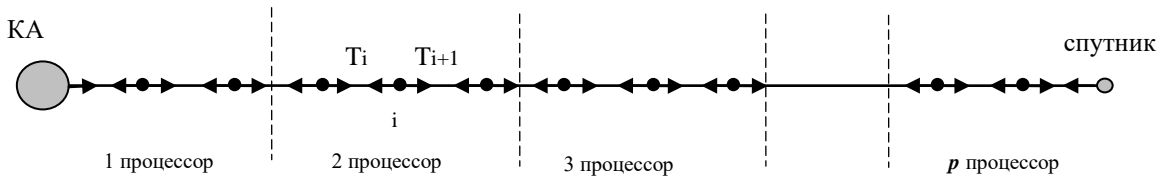


Рис. 2. Разбиение множества точек троса на сегменты.

Матрицу X разрежем по строкам на p матриц

$$X^k = \begin{pmatrix} x_{1,1}^k & x_{1,2}^k & \dots & x_{1,n}^k \\ x_{2,1}^k & x_{2,2}^k & \dots & x_{2,1}^k \\ \dots & \dots & \dots & \dots \\ x_{m,1}^k & x_{m,2}^k & \dots & x_{m,n}^k \end{pmatrix} \quad (11)$$

где $k = 1, \dots, p$, причем $X = \bigcup X^k$. По аналогии построим матрицу F^k . Теперь вычисление координат троса можно производить на каждом из p процессоров суперкомпьютера относительно независимо друг от друга. Зная индекс k , несложно установить соответствие между координатами матриц X^k и X .

Однако для работы параллельного алгоритма каждый из процессоров (сегмент троса) необходимо снабдить информацией о состоянии последней точки предшествующего сегмента и первой точки следующего сегмента. Для этого процессор с номером k должен иметь доступ к матрице D^k , содержащей фазовые координаты указанных точек:

$$D^k = \begin{pmatrix} x_{m,1}^{k-1} & x_{m,2}^{k-1} \\ x_{1,1}^{k+1} & x_{1,2}^{k+1} \end{pmatrix}. \quad (12)$$

В случае, когда параллельная программа создается для систем с общей памятью, данные для матрицы D^k можно получать непосредственно из матрицы X . Необходимо лишь обеспечить синхронизацию между процессорами, чтобы каждый из них получал координаты точек из соседнего сегмента, вычисленные другим процессором, в нужный момент времени. Для рассматриваемой задачи естественно выбрать механизм барьерной синхронизации, устанавливая барьер

после вычисления каждой матрицы K_i . Поскольку матрицы K_1, K_2, K_3, K_4 должны вычисляться последовательно, параллельный алгоритм можно представить в виде последовательности из четырех параллельных фаз вычислений. На каждой фазе параллельно вычисляется K_i . Процессор с номером k вычисляет строки матрицы K_i , соответствующие точкам его сегмента троса. Для этого он использует матрицы X^k и F^k . После окончания фазы расчета K_i процессы подходят к барьеру, чтобы получить новые фазовые координаты $\tilde{X}, \tilde{X}, \tilde{X}$ из матриц D^k , необходимые для расчета K_{i+1} .

При создании параллельной программы для систем с распределенной памятью после барьера процессы должны формировать матрицы D^k из сообщений от других процессов.

5. Графическая модель параллельного алгоритма

Изобразим параллельный алгоритм графически. Для этого воспользуемся нотацией технологии графосимволического программирования (ГСП) [6,7]. Технология ГСП позволяет описать алгоритм в виде набора диаграмм потоков управления, а затем автоматически скомпилировать и запустить программу, реализующую этот алгоритм. Диаграмма потоков управления представляет собой ориентированный граф, в котором вершины обозначают действия, выполняемые над данными некоторой предметной области, а дуги обозначают последовательность выполнения этих действий. Если из вершины исходят несколько дуг, то в зависимости от типа дуг может выполняться либо одна, либо одновременно несколько смежных вершин, в которые эти дуги ведут. Дуги различного типа обозначаются графически по-разному. Преимущество диаграмм потоков управления заключается в наглядном представлении последовательности развития вычислений в программе. Недостатком является отсутствие отображения зависимости по данным между вершинами. На рис. 3 представлен описанный выше параллельный алгоритм вычисления коэффициентов метода Рунге-Кутты.

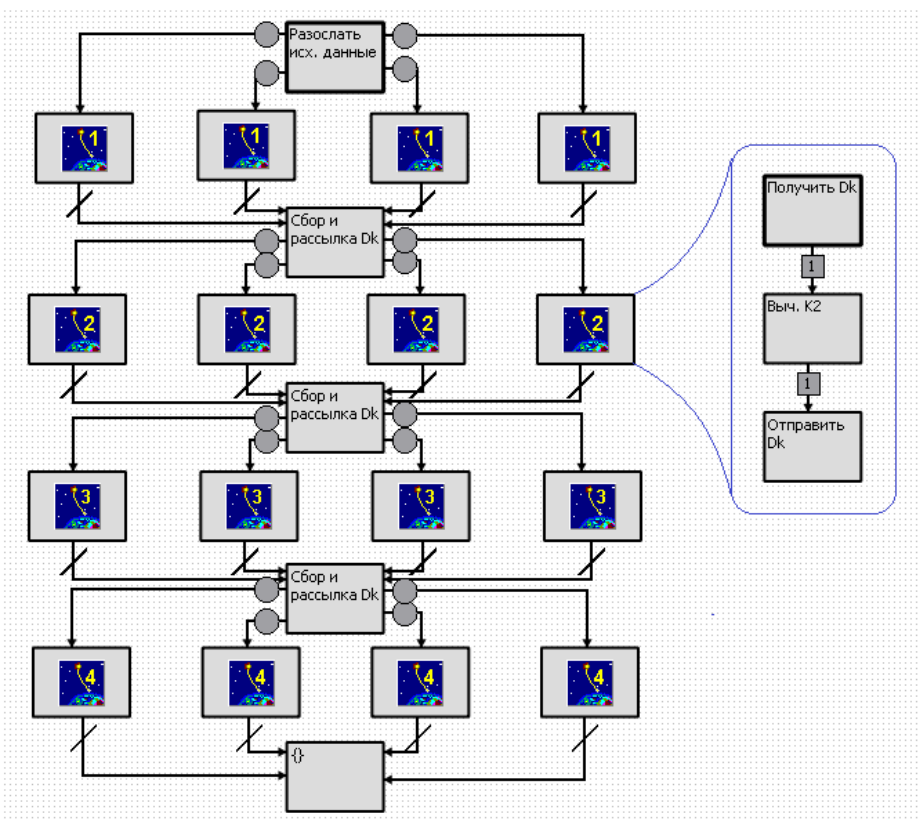


Рис. 3. Параллельный алгоритм вычисления коэффициентов метода Рунге-Кутты.

На рис. 3 дуги, начало которых помечено кружком, означают передачу управления в новый процесс. Дуги, начало которых помечено наклонной чертой, означают возврат управления из другого процесса. На каждой фазе вычислений используются четыре процесса для расчета различных сегментов троса. В технологии ГСП вершины могут помечаться текстом или изображениями. На рис. 3 некоторые вершины для наглядности помечены пиктограммами с упрощенным изображением космической тросовой системы и номером фазы вычислений. Эти вершины представляют собой последовательные подпрограммы вычисления K_i для отдельных сегментов троса. Таким образом, граф алгоритма на рис. 3 является иерархическим. Вершины с пиктограммами фактически представляют собой подграфы из трех вершин, содержимое которых приведено в правой части рисунка. Иерархическое построение графов алгоритмов в технологии

ГСП позволяет скрыть несущественные подробности реализации различных частей алгоритма, оставляя на графе только важные элементы с точки зрения понимания структуры алгоритма.

6. Оценка ускорения параллельного алгоритма численного моделирования движения тросовой системы

Оценим ускорение параллельной программы, реализующей предложенный алгоритм. Вычисление K_i для одной точки троса, в соответствии с формулой (9), сводится к вычислению значения функции f . Эта функция в рассматриваемой задаче включает в себя около 160 математических операций. Обозначим длительность выполнения одной математической операции через τ . Тогда вычисление K_i займет $\alpha = 160 \cdot \tau$. Продолжительность исполнения последовательной программы, которая вычисляет фазовые координаты для M точек на одном временном шаге, в соответствии с формулой (8) оценивается следующим выражением:

$$T_{\text{послед}}^1 \approx 4 \cdot \alpha \cdot M \quad (13)$$

Для моделирования движения в течение Nt шагов по времени суммарная продолжительность выполнения программы оценивается выражением

$$T_{\text{послед}}^{\text{сумм}} \approx 4 \cdot \alpha \cdot M \cdot Nt \quad (14)$$

Оценим продолжительность исполнения параллельной программы. В рассмотренном выше параллельном алгоритме на каждой фазе вычислений K_i вычисляется независимо на p процессорах. Продолжительность каждой фазы уменьшается в p раз по сравнению с последовательной программой. Общая длительность исполнения параллельной программы расчета фазовых координат M точек на одном временном шаге складывается из длительности выполнения четырех фаз вычислений и длительности обмена данными между сегментами:

$$T_{\text{парал}}^1 \approx \frac{4 \cdot \alpha \cdot M}{p} + T_{\text{обм}} \quad (15)$$

В программе для вычислительной системы с общей памятью обмен данными сводится к заполнению матриц D^k значениями фазовых координат из матрицы X . Эти операции представляют собой несколько обращений к оперативной памяти, что с учетом размеров матриц D^k пренебрежимо мало по сравнению с продолжительностью вычислений на каждой фазе. По этой причине временем обмена в программах для систем с общей памятью можно пренебречь. Расчет фазовых координат для различных шагов по времени выполняется последовательно, поэтому суммарная продолжительность выполнения параллельной программы оценивается следующим выражением:

$$T_{\text{парал}}^{\text{сумм}} \approx \frac{4 \cdot \alpha \cdot M}{p} \cdot Nt \quad (16)$$

Ускорение параллельной программы для систем с общей памятью составит

$$S = \frac{T_{\text{послед}}^{\text{сумм}}}{T_{\text{парал}}^{\text{сумм}}} = p \quad (17)$$

В системах с распределенной памятью каждый процесс в начале первой фазы вычислений должен получить исходные данные (матрицы X^k и F^k), а между фазами должен происходить обмен сообщениями, содержащими матрицы D^k . В качестве примера рассмотрим параллельную программу, использующую Message Passing Interface (MPI) для обмена данными между процессами. Главный процесс программы, осуществляющий ввод-вывод матриц X и F , должен разослать полосы этих матриц остальным процессам. Обозначим временные затраты на передачу 1 байта данных через β . Тогда на передачу X^k и F^k процессу с номером k главный процесс затратит

$$T_{\text{перед}}^{0,k} = \frac{\text{sizeof}(x_i) \cdot 4 \cdot M \cdot \beta}{p} + \frac{\text{sizeof}(x_i) \cdot 20 \cdot M \cdot \beta}{p} = \frac{24 \cdot \text{sizeof}(x_i) \cdot M \cdot \beta}{p} \quad (18)$$

В выражении (18) подразумевается, что каждая точка описывается четырьмя фазовыми координатами, а каждый элемент матрицы F^k содержит 20 параметров. Функция $sizeof(x_i)$ возвращает размер типа данных, которым представлены фазовые координаты.

Главный процесс рассылает матрицы X^k и F^k последовательно, поэтому суммарное время на рассылку исходных данных составит

$$T_{перед}^0 = \frac{24 \cdot sizeof(x_i) \cdot M \cdot \beta}{p} \cdot p = 24 \cdot sizeof(x_i) \cdot M \cdot \beta \quad (19)$$

Между фазами обмен данными происходит следующим образом. В начале каждой фазы главный процесс рассылает остальным процессам сообщения с матрицами D^k . В конце каждой фазы процессы отправляют главному процессу сообщения с новыми значениями координат, из которых он формирует матрицы D^k . Поскольку матрица D^k содержит 4 элемента, при последовательной рассылке суммарное время на обмен матрицами D^k в четырех фазах оценивается следующим выражением:

$$T_{перед}^{Dk} = (4 \cdot sizeof(x_i) \cdot \beta \cdot p) \cdot 4 \cdot 2 = 32 \cdot sizeof(x_i) \cdot \beta \cdot p \quad (20)$$

В выражении (20) подразумевается, что в каждой фазе вычислений пересылка значений для матриц D^k выполняется два раза: в начале фазы и в конце.

Поскольку $M \gg p$, $T_{перед}^0 \gg T_{перед}^{Dk}$. Оценим грубо абсолютные значения затрат времени на выполнение последовательной и параллельной программы. Пусть вычислительная система представляет собой кластер, узлы которого содержат по одному процессору с тактовой частотой 2Гц и соединяются между собой сетью Gigabit Ethernet. Тогда в качестве грубой оценки можно принять $\tau = 10^{-9}$ с, $\beta = 10^{-8}$ с. Пусть моделируется 1000 точек троса, координаты точек представлены числами с плавающей запятой двойной точности (тип double, sizeof(double)=8). С учетом принятых допущений получим:

$$T_{посл}^1 \approx 4 \cdot 10^{-9} \cdot 1000 = 4 \cdot 10^{-6} \quad (21)$$

$$\begin{aligned} T_{парал}^1 &\approx \frac{4 \cdot \alpha \cdot M}{p} + T_{обм} = \frac{4 \cdot \alpha \cdot M}{p} + T_{перед}^0 + T_{перед}^{Dk} = \frac{4 \cdot 10^{-6}}{p} + 24 \cdot 8 \cdot 10^3 \cdot 10^{-8} + 32 \cdot 8 \cdot 10^{-8} \cdot p = \\ &= \frac{4 \cdot 10^{-6}}{p} + 256 \cdot 10^{-8} \cdot p + 192 \cdot 10^{-5} \end{aligned} \quad (22)$$

Последнее слагаемое в выражении (22) существенно превышает значение выражения (21). Это означает, что затраты времени на рассылку исходных данных процессам, выполняющимся на узлах кластера, превышают затраты времени на выполнение последовательной программы расчета фазовых координат на одном временном шаге на одном узле. Следовательно, при создании параллельной программы для кластера необходимо выполнять рассылку исходных данных одновременно, до выполнения расчетов на первом временном шаге. При выполнении расчетов на последующих временных шагах процессы должны использовать локальные копии X^k и F^k , а обмениваться только значениями, необходимыми для построения матриц D^k .

7. Заключение

В работе рассмотрена задача численного моделирования движения космической тросовой системы на высокопроизводительных вычислительных системах. Указаны особенности задачи, влияющие на возможность распараллеливания вычислений. Предложен параллельный алгоритм реализации метода Рунге-Кутты, составленный с учетом особенностей решаемой задачи. Приведена оценка ускорения параллельной программы, реализующей алгоритм, при ее выполнении на системах с общей и распределенной памятью. Предложенный алгоритм описан в виде графической модели в технологии графо-символического программирования, позволяющей синтезировать параллельную программу на основе модели алгоритма. Графическая модель существенно упрощает анализ алгоритма, делает удобной его модификацию, способствует разработке новых алгоритмов и развитию исследований в области моделирования движения космических тросовых систем.

Благодарности

Исследование выполнено при финансовой поддержке РФФИ и Правительства Самарской области в рамках научного проекта 16-41-630637 «р_а».

Литература

- [1] Белецкий, В.В. Динамика космических тросовых систем / В.В. Белецкий, Е.М. Левин. — М.: Наука, 1990. — 336 с.
- [2] Эндрюс, Г. Основы многопоточного, параллельного и распределенного программирования / Г. Эндрюс. — М.: Вильямс, 2003. — 512 с.
- [3] Дигнат, Ф., Шилен, В. Управление колебаниями орбитальной тросовой системы // Прикладная математика и механика. — 2000. — Т. 64, вып. 5. — С. 747-754.
- [4] Заболотнов, Ю.М. Динамика движения капсулы с тросом на внеатмосферном участке спуска с орбиты / Ю.М. Заболотнов, Д.И. Фелелов // Известия СНЦ РАН. — 2006. — Т.8, № 3. — С. 841–848.
- [5] Crist, S. Cable Motion of a Spinning Spring-mass System in Orbit / S. Crist, J. Easley // J. of Spacecraft and Rockets. — 1969. — Vol.7, No.11. — P.819-824.
- [6] Коварцев, А.Н. Принципы построения технологии графосимволического программирования / А.Н. Коварцев, В.В. Жидченко, Д.А. Попова-Коварцева, П.В. Аболмасов // Труды II Международной научно-технической конференции Открытые семантические технологии проектирования интеллектуальных систем. — 2013. — С. 195-204.
- [7] Zhidchenko, V. Visual Parallel Programming as PaaS cloud service with Graph-Symbolic Programming Technology / V. Zhidchenko, D. Egorova // Proceedings of the Institute for System Programming. — 2015. Vol. 27 (Issue 3). — P. 47-56. DOI: 10.15514/ISPRAS-2015-27(3)-3.