

# Эффективность построения области достижимости летательного аппарата на графических процессорных устройствах

А.В. Быстров, А.П. Карпенко

МГТУ имени Н.Э. Баумана

Рассматривается задача построения области достижимости динамической системы и подход к решению этой задачи на основе метода «мультифиниша». Приводится схема распараллеливания задачи. Исследуется ускорение, полученное при расчете на графическом процессорном устройстве в случае наличия в области достижимости системы статических и динамических препятствий. Приводятся результаты параллельного решения задачи для высокоманевренного летательного аппарата.

## 1. Введение

Во многих приложениях, в которых модели исследуемых объектов представляют в виде обыкновенных дифференциальных уравнений (ОДУ), возникает задача построения области достижимости соответствующей динамической системы. В данной работе эта задача рассматривается в контексте проблемы обеспечения траекторной безопасности высокоманевренного летательного аппарата (ВМЛА) [1].

Важной особенностью задачи построения области достижимости ВМЛА является то, что ее приходится решать в режиме реального времени. При этом основные вычислительные затраты обусловлены затратами на интегрирование модельной системы ОДУ. Таким образом, задача быстрого построения области достижимости ВМЛА является актуальной.

Параллельные вычисления при численном построении некоторой аппроксимации области достижимости возможны на основе метода мультифиниша и нейросетевых методов. В работе использован метод мультифиниша, основная идея которого состоит в многократном численном интегрировании модельной системы ОДУ при различных допустимых управлениях [2]. Метод хорошо распараллеливается, как для MIMD, так и для SIMD многопроцессорных вычислительных систем [3, 4].

Работа продолжает исследования, начатые в работах [3, 4], и посвящена исследованию эффективности параллельного построения области достижимости на графических процессорных устройствах (ГПУ). Новым в работе является решение задачи в условиях наличия в области достижимости ЛА статических и динамических препятствий.

## 2. Постановка задачи

Рассмотрим динамическую систему

$$\dot{X} = F(t, X, U), \quad X(0) = X^0, \quad (1)$$

где  $X = X(t) = (x_1(t), x_2(t), \dots, x_n(t))^T$  –  $n$ -мерный вектор фазовых переменных системы,  $F = F(t, X, U) = (f_1(t, X, U), f_2(t, X, U), \dots, f_n(t, X, U))^T$  –  $n$ -мерная вектор-функция,  $U = U(t) = (u_1(t), u_2(t), \dots, u_m(t))^T$  –  $m$ -мерный вектор управлений,  $X^0 = (x_1^0, x_2^0, \dots, x_n^0)^T$  –  $n$ -мерный вектор начальных условий,  $t \in [0; T]$ . На вектор фазовых переменных  $X$  и вектор управления  $U$  наложены ограничения

$$X \in D_X, U \in D_U \subset L_U[0; T], \quad (2)$$

где  $L_U[0; T]$  – некоторое пространство  $m$ -мерных функций, определенных на интервале  $[0; T]$ , например, пространство функций, интегрируемых с квадратом на этом интервале.

Среди фазовых переменных  $x_1, x_2, \dots, x_n$  выделим  $\nu \leq n$  переменных. Не ограничивая общности, положим, что эти переменные образуют  $\nu$ -мерный вектор  $Y = Y(t) = (x_1(t), x_2(t), \dots, x_\nu(t))^T$ . Областью достижимости  $D_Y = D_Y(T, X^0)$  системы (1) назовем множество всех возможных значений вектора  $Y(t)$ , которые достигаются на решениях системы (1) при начальных условиях  $X^0$  и ограничениях (2).

### 3. Схема распараллеливания вычислений

Положим, что одну из границ множества достижимости формируют управления класса, в которых все компоненты вектора управления, кроме одной, постоянны, а компонента  $u_1(t)$ , например, имеет одну точку переключения:

$$u_1(t, t^S) = \begin{cases} +1, t \in [0; t^S], \\ -1, t \in (t^S; T], \end{cases} .$$

$$u_2(t) = \text{const}_1, \dots, u_m(t) = \text{const}_m$$

Здесь  $t^S \in [0; T]$  - момент времени, когда происходит переключение управления  $u_1(t)$ .

Покроем интервал  $[0; T]$  равномерной сеткой с шагом  $\Delta t^S = \frac{T}{M}$  и узлами  $t_i^S, i \in [1: M]$ .

Положим, что шаг  $\Delta t^S$  кратен шагу  $\Delta t$ , так что  $\Delta t^S = q\Delta t$ , где  $q = \frac{K}{M} \geq 1$  - целое число. Примем, что  $U_i = (u_1(t, t_i^S), u_2, \dots, u_m)^T$ .

При равномерной декомпозиции точек переключения схема распараллеливания имеет вид, представленный на рис. 1, где  $P_j$  -  $j$ -й процессор используемой параллельной вычислительной системе,  $N$  - число процессоров;  $j \in [1: N]$ .

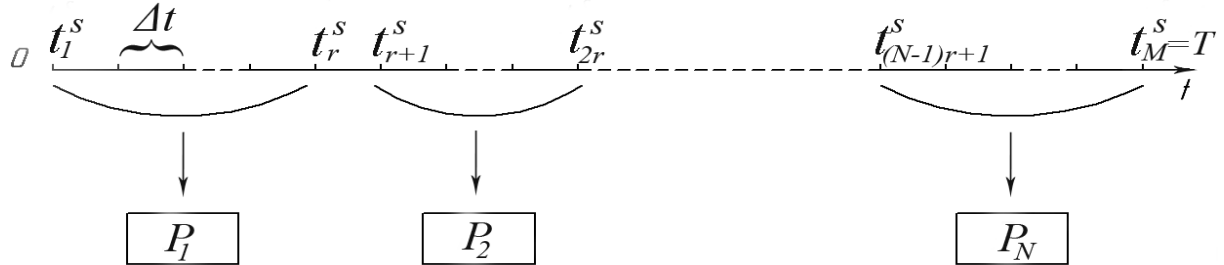


Рис. 1. Схема распараллеливания

В соответствии с этой схемой на процессоре  $P_1$  выполняется интегрирование системы ОДУ (1) при управлениях  $U_1, \dots, U_r$ , на процессоре  $P_2$  - при управлениях  $U_{r+1}, \dots, U_{2r}$  и т.д. до процессора  $P_N$ , который выполняет интегрирование при управлениях  $U_{(N-1)r+1}, \dots, U_M$ ;  $r$  - ближайшее целое большее  $M/N$ .

Вычисления на процессоре  $P_i, i \in [1: N]$  организуем следующим образом.

*Шаг 0* (этап «разгона»). Исходя из начальных условий  $X^0$ , выполняем интегрирование системы (1) при управлении  $(1, u_2, u_3, \dots, u_m)$  от начального момента до момента времени  $t_{ir-1}^S$  и сохраняем в памяти ЭВМ значения компонентов векторов  $X(t_{(i-1)r+1}^S) = X_{(i-1)r+1}^0$ ,  $X(t_{(i-1)r+2}^S) = X_{(i-1)r+2}^0, \dots, X(t_{ir-1}^S) = X_{ir-1}^0$ .

*Шаг 1.* Исходя из начальных условий  $X_{(i-1)r+1}^0$ , выполняем интегрирование той же системы при управлении  $(-1, u_2, u_3, \dots, u_m)$  от момента  $t_{(i-1)r+1}^S$  до момента времени  $T$ .

*Шаг 2.* Исходя из начальных условий  $X_{(i-1)r+2}^0$ , выполняем интегрирование при управлении  $(-1, u_2, u_3, \dots, u_m)$  от момента времени  $t_{(i-1)r+2}^S$  до момента времени  $T$ .

...

*Шаг r.* Исходя из начальных условий  $X_{ir-1}^0$ , выполняем интегрирование при управлении  $(-1, u_2, u_3, \dots, u_m)$  от момента времени  $t_{ir-1}^S$  до момента времени  $T$ .

Варианты оптимизации данной схеме рассмотрены в работе [2].

## 4 Математическая модель ВМЛА

Рассмотрим ВМЛА, движение центра масс которого в нормальной земной системе координат  $Oxuz$  описываются системой нелинейных дифференциальных уравнений

$$\left\{ \begin{array}{l} \dot{V} = g \cdot (n_T - \sin \Theta), \\ \dot{\Theta} = \frac{g}{V} (n \cdot \cos \gamma_c - \cos \Theta), \\ \dot{\Psi} = \frac{-g \cdot n \cdot \sin \gamma_c}{V \cdot \cos \Theta}, \\ \dot{X} = V \cdot \cos \Theta \cdot \cos \Psi, \\ \dot{Y} = V \cdot \sin \Theta, \\ \dot{Z} = -V \cdot \cos \Theta \cdot \sin \Psi \end{array} \right. \quad (3)$$

где  $V$  – скорость ВМЛА,  $\Theta$  – угол наклона траектории,  $\Psi$  – угол поворота траектории,  $Y$  – высота ВМЛА,  $n_T$  – тангенциальная перегрузка,  $n$  – нормальная перегрузка,  $\gamma_c$  – скоростной угол крена;  $g$  – ускорение свободного падения [1]. Нормированный вектор управления ВМЛА имеет вид

$$u = (u_1, u_2, \gamma_c), \quad |u_1| \leq 1; \quad |u_2| \leq 1; \quad |\gamma_c| \leq \pi,$$

где  $n^{\max} \cdot u_1 = n$ ,  $n^{\max} \cdot u_2 = n_T$ .

В работе [1] получена следующая структура управлений, приводящих ВМЛА на дальнюю, ближнюю и боковую границы области достижимости, соответственно:

$$\begin{aligned} u_{\partial} &= \left[ \begin{array}{l} u_1 = \pm 1 \\ u_2 = 1 \end{array} \right]_{t \in [0, t_1^-]} \cup \left[ \begin{array}{l} u_1 = 0 \\ u_2 = 1 \end{array} \right]_{t \in [t_1^-, T]} ; \\ u_{\partial} &= \left[ \begin{array}{l} u_1 = \mp 1 \\ u_2 = -1 \end{array} \right]_{t \in [0, t_1^+]} \cup \left[ \begin{array}{l} u_1 = \pm 1 \\ u_2 = -1 \end{array} \right]_{t \in [t_1^+, T]} ; \\ u_{\partial} &= \left[ \begin{array}{l} u_1 = \pm 1 \\ u_2 = -1 \end{array} \right]_{t \in [0, t_2^+]} \cup \left[ \begin{array}{l} u_1 = \pm 1 \\ u_2 = 1 \end{array} \right]_{t \in [t_2^+, T]} . \end{aligned}$$

Здесь  $t^-$  – момент времени перехода на особый участок управления  $n_{\text{особ}} = 0$ ,  $t_1^+$ ,  $t_2^+$  – моменты переключения знака управления нормальной перегрузкой и переключения знака управления тангенциальной перегрузкой соответственно.

## 5. Отображение алгоритма мультифиниша на архитектуру GPU

Покроем интервал времени  $[0; T]$  сеткой  $\Omega_u$  с  $n_u$  узлами, а область допустимых значений управления  $\gamma_c$  - сеткой  $\Omega_\gamma$  с  $m_\gamma$  узлами. Нити с идентификатором  $(i_u, i_\gamma)$  поставим в соответствие  $l$  соседних узлов сетки  $\Omega_u$ , соответствующих узлу  $i_\gamma$  сетки  $\Omega_\gamma$ . Говоря более строго, для данной границы области достижимости, например, указанная нить обрабатывает управление

$$\gamma_c = i_\gamma \frac{\gamma_c^{\max} - \gamma_c^{\min}}{m_\gamma}$$

и управления  $u_1$  или  $u_2$ , определяемые точками переключения

$$t_i = (i_u l + i) \frac{T}{n_u l}, \quad i \in [0 : (l-1)].$$

Структура GPU-программы имеет следующий вид:

- хост-процессор копирует в глобальную память GPU исходные данные (начальная скорость и координаты летательного аппарата);
- GPU реализует этап разгона, т.е. интегрирование системы (3) от начального момента времени до первой точки переключения  $t_k$ , соответствующей данной нити;
- GPU последовательно выполняет интегрирование той же системы при управлениях, соответствующих точкам переключения  $t_{k+i}$ ,  $i \in [0 : (l-1)]$ ;
- GPU осуществляет копирование координат полученных граничных точек на хост-процессор;
- хост-процессор записывает все результаты в файл и осуществляет визуализацию полученного множества точек.

## 5. Результаты исследования

### 5.1 Задача без препятствий

Исследование выполнено на хост-процессоре, имеющем следующие основные параметры: процессор - *Intel Core2 Duo CPU P8600 @ 2,40 ГГц*; ОЗУ - 4 ГБ. В качестве GPU использована видеокарта *NVIDIA GeForce 9650M GT*: число мультипроцессоров - 4; число ядер в мультипроцессоре - 8; общее число вычислительных ядер - 32; версия среды выполнения - *CUDA 3.20*; *Compute Capability* - 1.1; частота - 1.38 ГГц; объем глобальной памяти - 1054 Мб.

GPU-программа реализована на языке C с использованием технологии *CUDA*. Для визуализации границ области достижимости использована библиотека *OpenGL*. Эффективность параллельных вычислений оценивалось ускорением  $S$ , равным отношению времени решения задачи на хост-процессоре к времени ее решения на GPU.

Полученный вид области достижимости ВМЛА иллюстрирует рисунок 2. Здесь и далее, если не оговорено противное, результаты соответствуют пяти секундам полета ВМЛА со скоростью 300 м/с и начальными координатами (0, 0, 1000 м).

Выполнено широкое исследование эффективности параллельных вычислений при варьировании числа точек переключения  $n_u$ , числа  $m_\gamma$  узлов сетки  $\Omega_\gamma$ , а также времени полета ВМЛА  $T$ . Некоторые результаты исследования иллюстрирует рисунок 3.

Рисунок 3 показывает, что, как и следовало ожидать, ускорение возрастает с ростом числа точек переключения  $n_u$  и длительности полета  $T$ , т.е. с ростом вычислительной сложности задачи. Максимально достигнутое ускорение в условиях исследования лишь немного превышает 10. Меньшее ускорение объясняется, в первую очередь, потерями на копирование данных из оперативной памяти хост-процессора в глобальную память GPU и обратно.

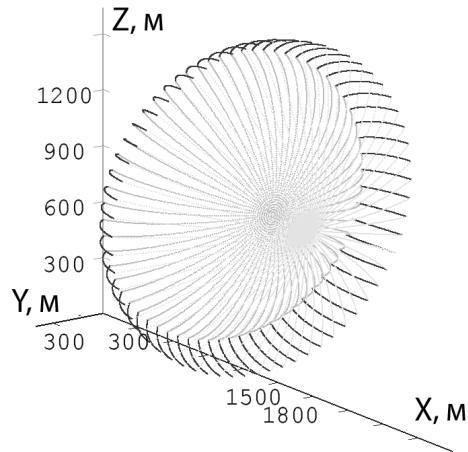


Рис. 2. Пример границ области достижимости ВМЛА

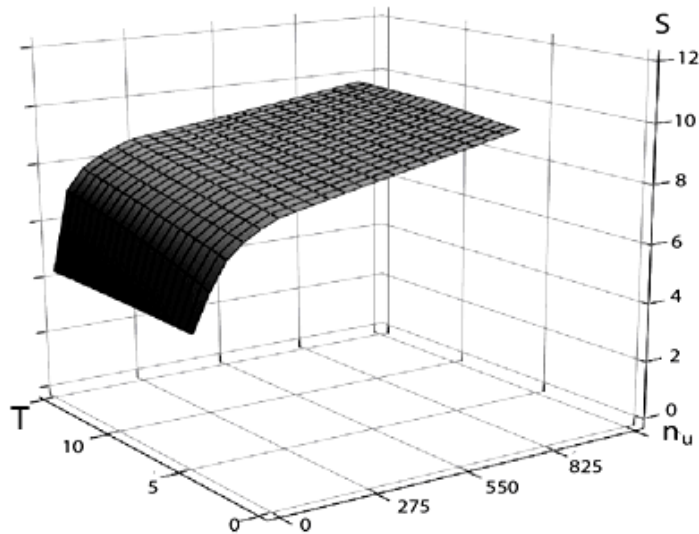


Рис. 3. Достигнутое ускорение в функции числа точек переключения  $n_u$  и длительности полета  $T$ :

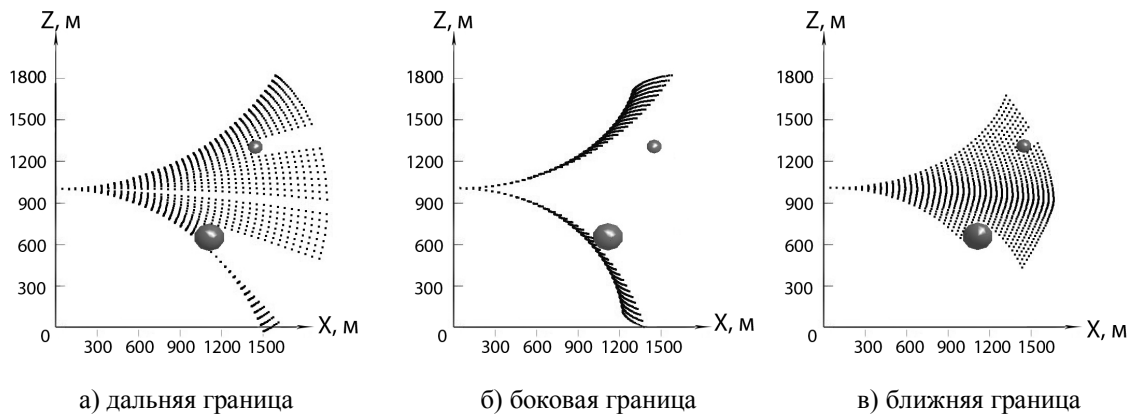
$$m_\gamma = 256$$

## 5.2 Стационарная задача с препятствиями

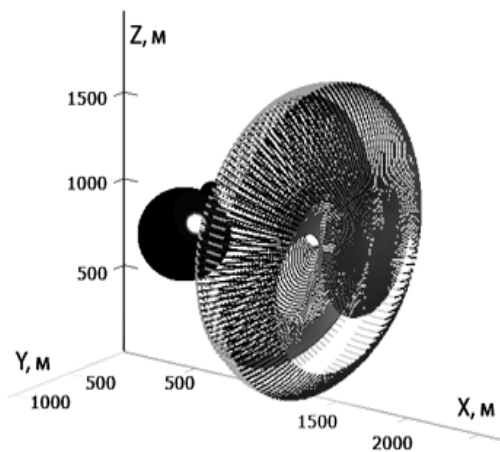
Положим, что в области достижимости ВМЛА находится одно или несколько стационарных сферических препятствий, имеющих известные радиусы и координаты центров. Наличие этих препятствий несколько меняет рассмотренный алгоритм построения области достижимости ВМЛА. На каждом шаге интегрирования в этом случае приходится проверять допустимость данной траектории и, если она пересекает то или иное препятствие, прекращать интегрирование. Характер соответствующей деформации области достижимости для двух препятствий в одной из координатных плоскостей иллюстрирует рисунок 4. Рисунок получен при следующих начальных условиях:

$$X = 0, \text{ м}; \quad Y = 0, \text{ м}; \quad Z = 1000, \text{ м}; \quad V = 330, \text{ м/с}; \quad \Theta = 0; \quad \Psi = 0.$$

Соответствующая трехмерная иллюстрация представлена на рисунке 5. При указанных начальных условиях получены также все остальные приведенные ниже результаты.

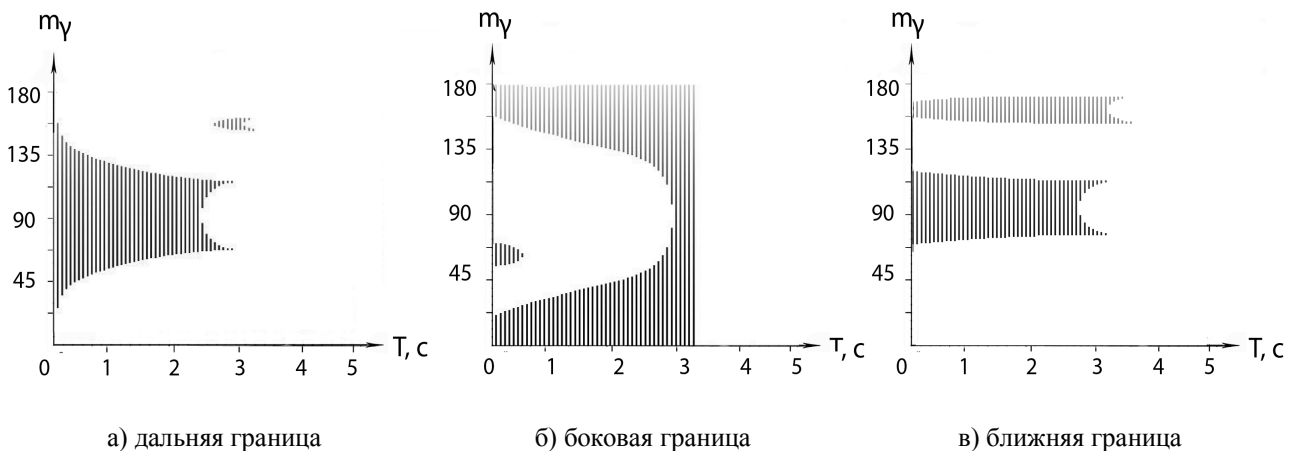


**Рис. 4.** Деформация области достижимости двумя препятствиями в плоскости



**Рис. 5.** Деформация области достижимости двумя препятствиями в пространстве

Одной из основных целей построения области достижимости ВМЛА при наличии препятствий является отыскание запрещенных управлений, т.е. управлений, приводящих к столкновению ЛА с препятствием. Для отыскания таких управлений выполнена еще одна модификация рассмотренной выше схемы параллельных вычислений, в соответствии с которой при обнаружении запрещенного управления его параметры заносятся в специальную таблицу. Пример визуализации такой таблицы для случая двух препятствий приведен на рисунке 6.



**Рис. 6.** Параметры запрещенных траекторий для случая двух препятствий

Исключение из рассмотрения запрещенных траекторий приводит к сокращению вычислительной сложности задачи и к тем большему сокращению, чем большее число траекторий яв-

ляются запрещенными. При этом время решения задачи уменьшается, но одновременно уменьшается и ускорение вычислений (рисунок 7.)

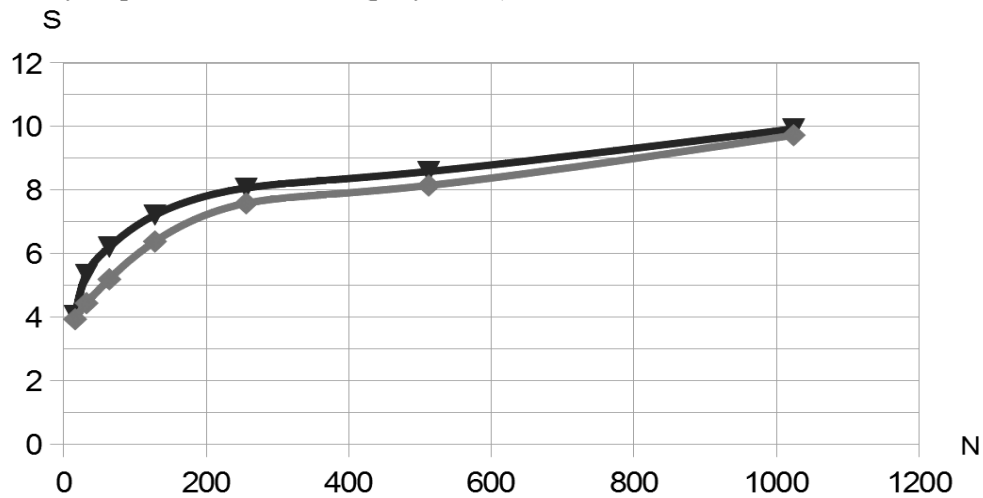


Рис. 7. Достигнутое ускорение при  $m_\gamma = 256$  и одном препятствии в функции числа точек переключения  $n_u$ : o - при наличии препятствия;  $\nabla$  - при отсутствии препятствия.

### 5.3 Динамическая задача с препятствиями

Выполнено исследование эффективности параллельных вычислений также для случая одного сферического препятствия, движущегося равномерно и прямолинейно, а также для случая аналогичного тяжелого препятствия, движущегося под действием силы тяжести. Характер деформации области достижимости ВМЛА в последнем случае иллюстрирует рисунок 8.

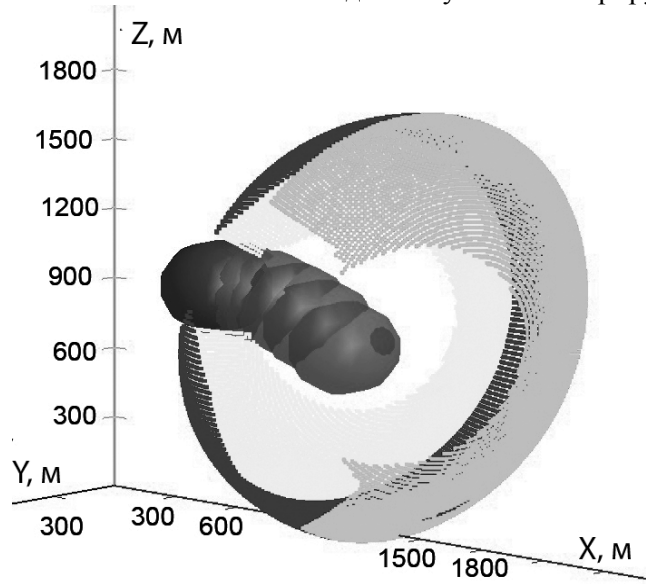


Рис. 8. Область достижимости ВМЛА для случая динамического сферического препятствия

Результаты исследования ускорения вычислений показывают результаты, близкие к представленным выше для стационарного препятствия.

## 6. Заключение

Результаты выполненного исследования показывают, что построения области достижимости ВМЛА на графической карте *Compute capability* 1.1 позволяет достигнуть примерно 11 кратного ускорения. Выполнено исследование причин, обуславливающих такое небольшое ускорение. Среди мер, по увеличению полученного ускорения можно назвать использование неравномерной декомпозиции точек переключения по потоковым процессорам, использование более новых графических карт (с *Compute capability* 2.0 и выше), применение нескольких графических процессорных устройств.

В целом, на основании выполненного исследования можно констатировать перспективность использования графических процессорных устройств для приближенного построения границ области достижимости ВМЛА в реальном времени, как в условия отсутствия в этой области препятствий, так и в случае наличия статических и динамических препятствий.

В развитие планируется исследование эффективности применения для решения указанной задачи нескольких графических процессорных устройств.

## Литература

1. Воронов, Е.М. Алгоритм оценки границ области достижимости летательного аппарата с учетом тяги / Е.М. Воронов, А.А. Карпунин // Вестник МГТУ. Сер. Приборостроение. - 2007. - №4(69). - с. 81-99.
2. Воронов, Е.М. Численные методы построения области достижимости динамической системы / Е.М. Воронов, А.П. Карпенко, О.Г. Козлова, В.А. Федин В.А. // Вестник МГТУ им. Н.Э. Баумана, серия «Приборостроение», 2010, №2 (79), с. 3-20.
3. Воронов, Е.М. Параллельное построение множества достижимости высокоманевренного летательного аппарата методом «мультифиниша» / Е.М. Воронов, А.П. Карпенко, В.А. Федин // Параллельные вычислительные технологии (ПаВТ'2010): Труды международной научной конференции (Уфа, 29 марта – 2 апреля 2010 г.) [Электронный ресурс] – Челябинск: Издательский центр ЮУрГУ, 2010. – с. 113-120.
4. Витюков, Ф.А. Построение области достижимости динамической системы на NVidia и AMD графических процессорах / Ф.А. Витюков, В.К. Домашнев, А.П. Карпенко, В.А. Федин В.А. // Научный сервис в сети Интернет: суперкомпьютерные центры и задачи Труды международной суперкомпьютерной конференции (21-26 сентября 2009 г., Новоросийск). – М.: Изд-во МГУ, 2010. – с.635-641.