

УДК 004.42

Дата подачи статьи: 19.05.17

DOI: 10.15827/0236-235X.119.524-528

2017. Т. 30. № 3. С. 524–528

## **ОПТИМИЗАЦИЯ ЗАДАЧИ ОБ ОПРЕДЕЛЕНИИ КОНФЛИКТОВ С ОПАСНЫМИ ЗОНАМИ ДВИЖЕНИЯ ЛЕТАТЕЛЬНЫХ АППАРАТОВ ДЛЯ ВЫПОЛНЕНИЯ НА INTEL XEON PHI**

*А.А. Рыбаков, к.ф.-м.н., ведущий научный сотрудник, rybakov@jscc.ru  
(Межведомственный суперкомпьютерный центр РАН – филиал ФНЦ НИИСИ РАН,  
Ленинский просп., 32а, г. Москва, 119334, Россия)*

Во время полета летательный аппарат генерирует вихревой спутный след. Данный след может представлять опасность для других участников воздушного движения, особенно в зонах, в которых одновременно находится большое количество летательных аппаратов. Поэтому задача моделирования вихревых следов и определения конфликтов с ними является крайне важной и востребованной.

В данной статье вихревой след рассматривается как совокупность опасных зон отдельных отрезков траектории движения. Опасная зона отдельного отрезка траектории представлена пучком сфер – множеством сфер с центрами на данном отрезке. Далее рассматривается прямой математический метод определения пересечения прямой и пучка сфер. Прямая представляет мгновенную траекторию движения собственного летательного аппарата, тогда как множество пучков сфер формирует полную опасную зону полета.

Так как количество отрезков траекторий в опасной зоне достаточно велико, целесообразно использование ускорителей для осуществления вычислений. В статье рассматриваются применение Intel Xeon Phi для данной задачи и использование специального приема ручной оптимизации исходного кода для того, чтобы добиться применения векторизации кода. Такой подход позволяет добиться пятикратного ускорения решения задачи при обходе около  $10^7$  отрезков движения внутри опасной зоны.

**Ключевые слова:** вихревой след, опасная зона движения летательного аппарата, определение конфликтов, Intel Xeon Phi, векторизация.

Во время движения летательный аппарат (ЛА) создает вихревые возмущения, которые остаются вдоль траектории его полета. Совокупность этих возмущений формирует турбулентный (вихревой) след ЛА [1], который со временем эволюционирует и в конечном итоге разрушается [2]. Пока вихревой след существует, он может представлять опасность для других участников воздушного движения при попадании их в данный след [3]. Особенно это критично в местах скопления ЛА, где на небольшом участке пространства может находиться сразу большое количество вихревых следов (аэродромы) [4]. Поэтому необходимо осуществлять постоянный мониторинг вихревых следов вплоть до их разрушения и оценивать опасность их влияния на собственный ЛА [5]. В данной статье рассматривается задача определения потенциальных конфликтов с множеством вихревых следов ЛА. Приводится алгоритм решения данной задачи, а также рассматривается его реализация на вычислителе Intel Xeon Phi.

### **Постановка задачи**

Траекторию движения ЛА зададим набором точек в пространстве, последовательно соединенных отрезками. Будем считать, что в данных точках известны характеристики ЛА, в том числе и интенсивность создаваемых им вихревых возмущений, а также то, что опасная зона одной конкретной точки представляет собой шар с центром в этой точке. Точку в трехмерном пространстве зададим радиус-вектором, направленным из начала координат в эту

точку, а сферу – ее центром и радиусом следующим образом:  $S(\bar{C}, R) = \left\{ \bar{P} \mid \left| \bar{P} - \bar{C} \right|^2 = R^2 \right\}$ .

Примем, что опасная зона отрезка является объединением опасных зон всех его точек, где радиус опасной зоны точки отрезка меняется линейно от одного его конца до другого. Для задания опасной зоны отрезка введем понятие пучка сфер, опирающегося на две различные сферы. Пусть заданы две сферы:  $S_0(\bar{C}_0, R_0)$  и  $S_1(\bar{C}_1, R_1)$ .

Тогда пучком сфер, опирающимся на них, назовем такое множество сфер, центры и радиусы которых определяются следующим образом:

$$\begin{cases} \bar{C}(\alpha) = \bar{C}_0 + \alpha(\bar{C}_1 - \bar{C}_0) = \bar{C}_0 + \alpha\Delta\bar{C}, \\ R(\alpha) = R_0 + \alpha(R_1 - R_0) = R_0 + \alpha\Delta R, \end{cases}$$

где параметр  $\alpha$  изменяется от 0 до 1. Нетрудно заметить, что опасная зона отрезка представляет собой область, ограниченную опасными зонами его концов, являющимися шарами, и множеством общих внешних касательных данных шаров. Кроме того, отметим, что опасная зона отрезка является выпуклым телом. Опасную зону всей траектории движения ЛА будем определять как объединение опасных зон всех отрезков его движения

На рисунке 1 показан пример траектории движения ЛА, закрашенными кругами обозначены опасные зоны точек, в которых известны характеристики ЛА, также очерчена опасная зона всей траектории движения.

Итак, в задаче обнаружения возможных конфликтов с опасными зонами траекторий движения

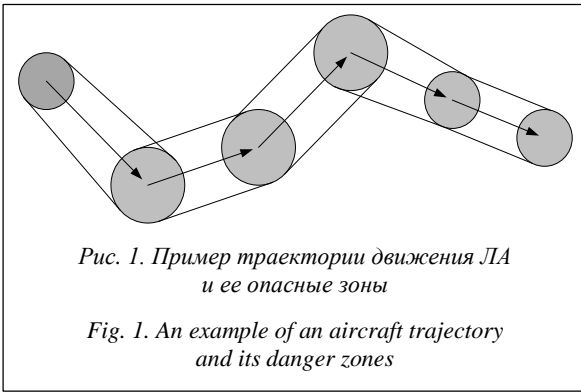


Рис. 1. Пример траектории движения ЛА и ее опасные зоны

Fig. 1. An example of an aircraft trajectory and its danger zones

в местах скопления ЛА задаются множество траекторий движения различных ЛА, а также положение и вектор мгновенной скорости собственного ЛА, для которого нужно определить конфликт. Положение и вектор скорости собственного ЛА определяют луч, состоящий из множества точек  $\overline{P}(t) = \overline{P}_0 + t\overline{V}$  при  $t \geq 0$ , для которого нужно определить, пересекает ли он опасную зону траектории движения хотя бы одного ЛА. В случае конфликта требуется найти точку первого пересечения для определения уровня угрозы, то есть времени, оставшегося до пересечения опасной зоны.

### Решение

Для решения поставленной задачи необходимо решить задачу пересечения прямой с пучком сфер. Сначала рассмотрим пересечение прямой и отдельной сферы. Без ограничения общности будем считать, что собственный ЛА находится в начале координат (этого всегда можно добиться с помощью параллельного переноса). Пересечение прямой со сферой находится из следующей системы уравнений:

$$\begin{cases} \overline{P} = t\overline{V}, \\ |\overline{P} - \overline{C}|^2 = R^2. \end{cases}$$

Данная система уравнений преобразуется в квадратное уравнение относительно  $t$ , дискриминант которого равен  $4((\overline{C} \cdot \overline{V})^2 - |\overline{V}|^2 (|\overline{C}|^2 - R^2))$ .

Так как значение дискриминанта может быть вычислено непосредственно, дальнейшее нахождение точек пересечения не представляет труда (если он неотрицательный).

Перейдем к поиску точек пересечения прямой и пучка сфер. Для каждой конкретной сферы из пучка можно выписать условие наличия точек пересечения:

$$(\overline{C}(\alpha) \cdot \overline{V})^2 - |\overline{V}|^2 (|\overline{C}(\alpha)|^2 - R(\alpha)^2) \geq 0.$$

Подставив конкретные выражения для центра и радиуса сферы и выполнив необходимые преобразования, получим квадратное неравенство относительно параметра  $\alpha$ :  $A_2\alpha^2 + 2A_1\alpha + A_0 \geq 0$ , где

$$\begin{cases} A_2 = (\overline{\Delta C} \cdot \overline{V})^2 + |\overline{V}|^2 (\Delta R^2 - |\overline{\Delta C}|^2), \\ A_1 = (\overline{C}_0 \cdot \overline{V})(\overline{\Delta C} \cdot \overline{V}) + |\overline{V}|^2 (R_0 \Delta R - (\overline{C}_0 \cdot \overline{\Delta C})), \\ A_0 = (\overline{C}_0 \cdot \overline{V})^2 + |\overline{V}|^2 (R_0^2 - |\overline{C}_0|^2). \end{cases}$$

Полученное неравенство можно решить относительно  $\alpha$ , так как все коэффициенты вычисляются непосредственно. Нас интересует решение данного неравенства только на отрезке  $[0, 1]$ , оно представляет собой либо пустое множество, либо отрезок.

Если на отрезке  $[0, 1]$  приведенное неравенство не имеет решения, прямая не пересекает пучок сфер. Допустим, на отрезке  $[0, 1]$  данное неравенство имеет решение  $[\alpha_0, \alpha_1] \subseteq [0, 1]$ .

Для произвольного значения  $\alpha$  из данного отрезка корни уравнения

$$|\overline{V}|^2 t^2 - 2(\overline{C}(\alpha) \cdot \overline{V})t + (|\overline{C}(\alpha)|^2 - R(\alpha)^2) = 0$$

существуют и выражаются следующим образом:

$$t_{1,2}(\alpha) = \frac{(\overline{C}(\alpha) \cdot \overline{V}) \pm \sqrt{A_2\alpha^2 + 2A_1\alpha + A_0}}{|\overline{V}|^2}.$$

Искомые точки пересечения прямой с пучком сфер соответствуют минимальному и максимальному значениям параметра  $t(\alpha)$ , достигаемым на отрезке  $[\alpha_0, \alpha_1]$ . Максимальное и минимальное значения  $t(\alpha)$  могут достигаться либо на концах отрезка, либо в точках локального экстремума. Для точки локального экстремума должно быть выполнено соотношение  $t'_{1,2}(\alpha) = 0$  или в явном виде

$$(\overline{\Delta C} \cdot \overline{V}) \pm \frac{A_2\alpha + A_1}{\sqrt{A_2\alpha^2 + 2A_1\alpha + A_0}} = 0,$$

$$\begin{cases} q(A_2\alpha^2 + 2A_1\alpha + A_0) = (A_2\alpha + A_1)^2, \\ q = (\overline{\Delta C} \cdot \overline{V})^2. \end{cases}$$

После преобразования получим следующее квадратное уравнение относительно  $\alpha$ :

$$A_2(q - A_2)\alpha^2 + 2A_1(q - A_2)\alpha + (qA_0 - A_1^2) = 0,$$

решая которое, найдем потенциальные точки локальных экстремумов  $\alpha_1, \alpha_2$ .

Данные точки нужно учитывать только в случае их попадания в отрезок  $[\alpha_0, \alpha_1]$ . В общем случае значения находятся по четырем значениям  $\alpha$ :

$$\begin{cases} t_1 = \min \{t_1(\alpha_0), t_1(\alpha_1), t_1(\alpha_1), t_1(\alpha_2)\}, \\ t_2 = \max \{t_2(\alpha_0), t_2(\alpha_1), t_2(\alpha_1), t_2(\alpha_2)\}. \end{cases}$$

По найденным значениям непосредственно находятся точки пересечения прямой с пучком сфер. Для задачи, рассматриваемой в статье, необходимо получить ответ о пересечении с пучком сфер не на всей прямой, а на луче, соответствующем неотрицательным  $t$ . Конечно, возможен случай, когда собственный ЛА уже находится в опасной зоне и пересечение с данной опасной зоной

произошло при  $t < 0$ , но данный случай легко сводится к общему.

### Особенности решения

Рассмотрим подробнее решение представленного ранее квадратного неравенства  $A_2\alpha^2 + 2A_1\alpha + A_0 \geq 0$ .

Утверждалось, что решением данного неравенства на отрезке  $[0, 1]$  является либо пустое множество, либо отрезок. Единственным потенциально возможным вариантом, когда это не так, является вариант, при котором  $A_2 > 0$  и оба значения  $\alpha$ , при которых неравенство обращается в равенство, находятся внутри отрезка  $[0, 1]$ . Это означает, что прямая пересекает сферы  $S_0$  и  $S_1$ , однако не пересекает хотя бы одну другую сферу из пучка. Но это невозможно в силу выпуклости пучка сфер, а значит, этот случай исключен.

Теперь рассмотрим подробнее случай  $A_2 < 0$ :

$$(\overline{\Delta C} \cdot \overline{V})^2 + |\overline{V}|^2 (\Delta R^2 - |\overline{\Delta C}|^2) < 0.$$

Раскрыв скалярное произведение векторов и выполнив необходимые преобразования, получим следующее условие на угол между скоростью собственного ЛА и отрезком, содержащим центры сфер пучка:  $|\sin(\overline{\Delta C}, \overline{V})| > \frac{|\Delta R|}{|\overline{\Delta C}|}$ , где  $|\Delta R|/|\overline{\Delta C}|$

представляет собой синус угла раствора пучка сфер. Заметим, что данный угол раствора всегда очень мал, так как характеристики ЛА меняются медленно во время движения и  $|\Delta R|$  близко к нулю. Таким образом, случай  $A_2 < 0$  является наиболее частым и данное условие выполняется в подавляющем большинстве случаев (рис. 2).

### Программная реализация и оптимизация

Опишем программную реализацию функции, которая по набору пучков сфер выдает координаты точек пересечения с ними. Для простоты будем

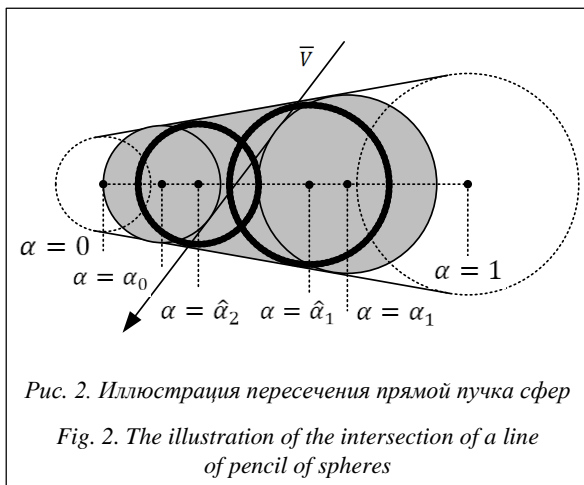


Рис. 2. Иллюстрация пересечения прямой пучка сфер  
 Fig. 2. The illustration of the intersection of a line of pencil of spheres

считать, что задана только одна траектория движения ЛА. Входными данными функции являются четыре массива: три массива с координатами точек, последовательно соединенных отрезками  $(x, y, z)$ , и один массив с радиусами опасных зон этих точек  $(r)$ . Также на вход подается вектор скорости собственного ЛА. Тогда алгоритм поиска точек пересечения с одним пучком сфер можно записать в виде следующей последовательности шагов.

1. Вычислить коэффициенты  $A_2, A_1, A_0$ , фигурирующие в квадратном неравенстве.
2. Если  $A_2 > 0$ , решить квадратное неравенство на отрезке  $[0, 1]$  и перейти к п. 5, иначе перейти к п. 3.
3. Если  $A_2 < 0$ , решить квадратное неравенство на отрезке  $[0, 1]$  и перейти к п. 5, иначе перейти к п. 4.
4. Так как  $A_2 = 0$ , решить линейное неравенство на отрезке  $[0, 1]$  и перейти к п. 5.
5. Если решение неравенства из пп. 2–4 на отрезке  $[0, 1]$  пусто, перейти к следующему пучку сфер, иначе перейти к п. 6.
6. На отрезке  $[\alpha_0, \alpha_1]$ , найденном в пп. 2–4, найти точки, в которых значения  $t_{1,2}$  достигают своих экстремумов.
7. По значениям  $\alpha_0, \alpha_1, \alpha_1, \alpha_2$  в общем случае найти моменты времени пересечения прямой с пучком сфер.

Данный алгоритм был реализован на языке C, а исполняемые коды опробованы на ускорителях Intel Xeon Phi 7110X [6], которые входят в состав находящегося в МСЦ РАН вычислительного комплекса МВС-10П [7]. Однако полученные результаты выявили неэффективное использование возможностей ускорителя. Основной проблемой стало наличие в алгоритме довольно сложного управления. В частности, разбор различных случаев при решении квадратного неравенства и при нахождении точек экстремумов  $\alpha_1, \alpha_2$ . Такое сложное управление привело к невозможности эффективного применения векторизации в цикле обработки пучков сфер [8].

Так как оптимизирующий компилятор [9] из состава Intel Parallel Studio, который использовался для компиляции кода, не смог подготовить эффективный исполняемый код, для устранения недостатков в эффективности было принято решение модифицировать код таким образом, чтобы удалить из цикла маловероятные ветви исполнения [10]. Как было показано выше, в подавляющем большинстве случаев при решении квадратного неравенства выполняется условие  $A_2 < 0$ . Однако даже в этом случае решение квадратного неравенства на отрезке  $[0, 1]$  почти всегда представляет собой пустое множество. Переписав условие отсутствия пересечений с пучком сфер, в данном случае получим выражение, которое истинно с вероятностью, близкой к единице:

$$(A_2 < 0) \wedge (m > 0) \wedge (A_1^2 - A_2 A_0 < m^2),$$

где  $m = \max(A_1 + A_2, -A_1)$ . Если обозначить это вероятное условие через *cond*, можно переписать основную часть цикла обработки пучков сфер в следующем виде:

```
for (<цикл по всем пучкам сфер>)
{
  <вычисления>

  if (cond)
  {
    continue; // prob. ~100%
  }
  else
  {
    <маловероятные вычисления>
  }
}
```

Данный цикл можно расщепить на два цикла, используя дополнительный массив, содержащий значения условий *cond*. При этом получим следующий результат:

```
for (<цикл по всем пучкам сфер>)
{
  <вычисления>
  tmp[i] = cond;
}

for (<цикл по всем пучкам сфер>)
{
  if (!tmp[i])
  {
    <маловероятные вычисления>
  }
}
```

В оптимизированном варианте основная часть вычислений приходится на первый цикл, при этом он не содержит сложного управления, а значит, подходит для векторизации. Второй цикл, конечно, неэффективно исполняется на ускорителе, но его

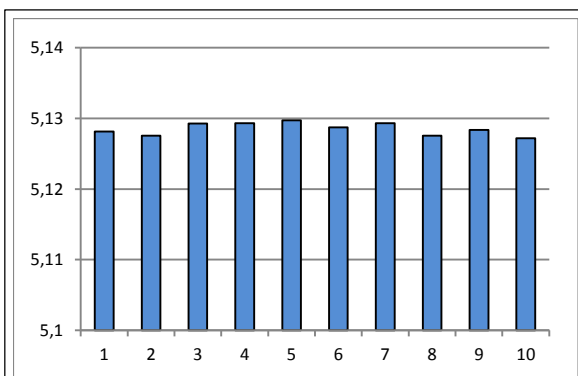


Рис. 3. Результаты ускорения оптимизированного варианта определения конфликта с опасными зонами

Fig. 3. The results of the acceleration of the optimized version of dangerous zones conflict definition

время исполнения настолько мало, что им можно пренебречь.

Были проведены сравнительные запуски с моделированием случайных траекторий движения ЛА для неоптимизированного и оптимизированного вариантов и получены стабильные результаты, которые демонстрируют ускорение вычислений за счет векторизации более чем в 5 раз по сравнению с неоптимизированным вариантом.

На рисунке 3 представлены результаты запусков. По горизонтальной оси отмечены 10 запусков, каждый из которых состоял из обработки  $10^7$  пучков сфер. По вертикали указано ускорение, полученное при использовании векторизованного варианта по сравнению с исходным.

### Заключение

При решении реальных практических задач с применением современных аппаратных вычислительных ресурсов сложно ожидать эффективного исполнения программы даже при использовании высококачественного компилятора, способного проводить оптимизацию кода. Зачастую задача требует вмешательства исследователя, способного определить особенности алгоритма и должным образом использовать их. Часто небольшие изменения логики вычислений, не влияющие на результат, но использующие особенности алгоритма, способны привести к кратному повышению производительности.

### Литература

1. Аубакиров Т.О., Желанников А.И., Иванов П.Е., Ништ М.И. Спутные следы и их воздействие на летательные аппараты. Моделирование на ЭВМ. Алматы: Гылым, 1999. 280 с.
2. Вышинский В.В., Судаков Г.Г. Вихревой след самолета в турбулентной атмосфере // Тр. ЦАГИ. 2006. Вып. 2667. 155 с.
3. Бабкин В.И., Белоцерковский А.С., Турчак Л.И., Баранов Н.А., Замятин А.И., Каневский М.И., Морозов В.В., Паскунов И.В., Чижов Н.Ю. Системы обеспечения вихревой безопасности полетов летательных аппаратов. М.: Наука, 2008. 373 с.
4. Бурлуцкий С.Г. Вопросы обеспечения вихревой безопасности аэропортов // Системный анализ и логистика. 2014. № 11. С. 37–40.
5. Кудрявцев Н.С. Оценка эффективности систем обеспечения вихревой безопасности полетов // Естественные и математические науки в современном мире. 2016. № 11. С. 31–37.
6. Jeffers J., Reinders J. Intel Xeon Phi coprocessor high performance programming. Morgan Kaufmann Publ., 2013, 432 p.
7. Описание интерфейса пользователя, предназначенного для работы с интеловской гибридной архитектурой суперЭВМ... URL: <http://www.jscc.ru/informat/MVS-10PInter.pdf> (дата обращения: 15.05.2017).
8. Шабанов Б.М., Телегин П.Н., Аладышев О.С. Особенности использования многоядерных процессоров // Программные продукты и системы. 2008. № 2. С. 7–9.
9. Aho A.V., Lam M.S., Sethi R., Ullman J.D. Compilers: principles, techniques, and tools (2nd Ed.). Pearson Education Inc. Publ., 2006, 1038 p.
10. Allen R., Kennedy K. Optimizing compilers for modern architectures. Morgan Kaufmann Publ., 2001, 790 p.

**OPTIMIZATION OF THE PROBLEM OF CONFLICT DETECTION WITH DANGEROUS AIRCRAFT  
MOVEMENT AREAS TO EXECUTE ON INTEL XEON PHI****A.A. Rybakov**<sup>1</sup>, Ph.D. (Physics and Mathematics), Leading Researcher, rybakov@jssc.ru<sup>1</sup> Joint Supercomputer Center of RAS, Lenin Ave. 32a, Moscow, 119991, Russian Federation

**Abstract.** An aircraft generates a wake vortex during its flight. This wake might be dangerous for other air traffic participants, especially in zones with a large number of aircrafts. So, the problem of vortex wake modeling and conflicts detection is very important and in demand.

The article considers a vortex wake as a set of dangerous zones in separate traffic segments. A dangerous zone of separate traffic segment is a pencil of spheres – a set of spheres with centers on this segment. Next, we consider the direct mathematical method of detecting an intersection of a line and a pencil of spheres. The line is an immediate trajectory of own aircraft, while a pencil of spheres forms the whole dangerous flight zone.

Since the number of traffic segments in a dangerous zone is large enough, it is appropriate to use accelerators in such calculations. The paper considers using Intel Xeon Phi coprocessors for this task and applying some manual sourcecode optimization to enforce execution code vectorization. This approach leads to five-time acceleration of calculation for 10<sup>7</sup> traffic segments in a dangerous zone.

**Keywords:** vortex wake, aircraft trajectory dangerous zone, conflicts detection, Intel Xeon Phi, vectorization.

**References**

1. Aubakirov T.O., Zhelannikov A.I., Ivanov P.E., Nisht M.I. *Sputnye sledy i ikh vozdeystvie na letatelnye apparaty. Modelirovanie na EVM* [Companion Tracks and Their Impact on an Aircraft. Computer Simulation]. Almaty, 1999, 280 p.
2. Vyshinsky V.V., Sudakov G.G. Aircraft vortex trail in turbulent atmosphere. *Trudy TsAGI* [TsAGI Science Jour.]. 2006, iss. 2667, 155 p. (in Russ.).
3. Babkin V.I., Belotserkovsky A.S., Turchak L.I., Baranov N.A., Zamyatin A.I., Kanevsky M.I., Morozov V.V., Paskunov I.V., Chizhov N.Yu. *Sistemy obespecheniya vikhrevoy bezopasnosti poletov letatelnykh apparatov* [Vortex Safety Systems for Aircrafts]. Moscow, Nauka Publ., 2008, 373 p.
4. Burlutsky S.G. Vortex security for airports. *Sistemny analiz i logistika* [Int. Jour. of System Analysis and Logistics Sciences]. 2014, iss. 11, pp. 37–40 (in Russ.).
5. Kudryavtsev N.S. Evaluation of vortex safety systems effectiveness. *Estestvennye i matematicheskie nauki v sovremen-nom mire* [Natural and Mathematical Sciences in the Modern World]. 2016, no. 11 (46), pp. 31–37 (in Russ.).
6. Jeffers J., Reinders J. *Intel Xeon Phi Coprocessor High Performance Programming*. Morgan Kaufmann Publ., 2013, 432 p.
7. *Opisanie interfeysa polzovatelya, prednaznachennogo dlya raboty s intelovskoy gibridnoy arkhitekturoy superEVM...* [A Description of a User Interface Designed to Work with the Intel Hybrid Supercomputer (SC), where Intel Xeon Phi Coprocessors are Used Together with Intel Xeon Processors]. Available at: <http://www.jssc.ru/informat/MVS-10PInter.pdf> (accessed May 15, 2017).
8. Shabanov B.M., Telegin P.N., Aladyshev O.S. Features of using multi-core processors. *Programmnye produkty i sistemy* [Software & Systems]. 2008, no. 2, pp. 7–9 (in Russ.).
9. Aho A.V., Lam M.S., Sethi R., Ullman J.D. *Compilers: Principles, Techniques, and Tools*. 2nd ed., Pearson Education Inc. Publ., 2006, 1038 p.
10. Allen R., Kennedy K. *Optimizing Compilers for Modern Architectures*. Morgan Kaufmann Publ., 2001, 790 p.

**Примеры библиографического описания статьи**

1. Рыбаков А.А. Оптимизация задачи об определении конфликтов с опасными зонами движения летательных аппаратов для выполнения на Intel Xeon Phi // Программные продукты и системы. 2017. Т. 30. № 3. С. 524–528. DOI: 10.15827/0236-235X.119.524-528.
2. Rybakov A.A. Optimization of the problem of conflict detection with dangerous aircraft movement areas to execute on Intel Xeon Phi. *Programmnye produkty i sistemy* [Software & Systems]. 2017, vol. 30, no. 3, pp. 524–528 (in Russ.). DOI: 10.15827/0236-235X.119.524-528.