

УДК 519.683.8

М. Е. Боднюк¹, А. Н. Знайдюк¹, Л. А. Байкова¹

¹ Иркутский государственный университет путей сообщения, г. Иркутск, Российская Федерация

ПРИМЕНЕНИЕ МАТЕМАТИКИ В ПРОГРАММИРОВАНИИ НА ПРИМЕРЕ ВЕКТОРНОЙ АЛГЕБРЫ

Аннотация. В статье описывается роль математических методов решения задач в программном обеспечении цифровых систем. Рассмотрена задача, в которой применены знания векторной алгебры в программно-аппаратном комплексе. Решение задачи представлено с помощью применения программного кода.

Ключевые слова: математика, векторная алгебра, программирование, автопилот.

М. Е. Bodnyuk¹, А. N. Znaydiuk¹, L. A. Baykova¹

¹ Irkutsk State University of Railway Transport, Irkutsk, Russian Federation

APPLICATION OF MATHEMATICS IN PROGRAMMING USING THE EXAMPLE OF VECTOR ALGEBRA

Abstract. The article describes the role of mathematical methods for solving problems in the software of digital systems. Several problems are considered, in which the knowledge of vector algebra in the software and hardware complex is applied. In these examples, the authors solve the problem using the application of program code.

Keywords: Mathematics, vector algebra, programming, autopilot.

Введение

Математика и программирование - две разные сферы знаний, однако они весьма тесно связаны. Математика широко используется в программном обеспечении различных устройств и систем, для математического моделирования природных явлений, емких и монотонных расчетов, проектирования и функционирования искусственного интеллекта, и т.п. Как правило при выполнении расчетов задействуются многочисленные операции из разных областей математики. Помимо формул, с помощью которых решаются задачи, математика дает вариативность этих самых решений, что в свою очередь дает разную эффективность решений. Программирование позволяет вычислять большое количество математических операций с постоянно меняющимися переменными.

Для любого явления в жизни можно написать программный код, который смоделирует это явление. В данном коде, как правило, есть две части: внешняя и внутренняя. Внешняя часть служит как посредник между данными и внутренней частью. Внутренняя же часть представляет из себя комплект математических формул, с помощью которых функционирует программа.

Конечно, некоторые читатели скажут: - «Зачем разбираться в основах программирования и писать код для вычисления, если можно решить задачу на листке бумаги». И будут отчасти правы. Некоторые задачи действительно проще решить вручную. Однако такие программы составляют, когда требуется работать с постоянно обновляющимися данными, громоздкими и муторными вычислениями, а также результат должен быть найден в течение доли секунды.

Примером системы, в которой соблюдение данных требований должно быть обязательным, является программно-аппаратный комплекс "автопилот".

В этой статье для показательного примера из всей математики мы возьмем только векторную алгебру.

История вопроса

Первый автопилот появился в 1912 году. Он работал с помощью гироскопа и помогал предотвращать излишние смещение направления движения вдоль осей самолета. Эта система была неэффективна в плохую погоду. Современные автопилоты устроены куда сложнее и надежнее. В них встроены GPS, что позволяет им преодолеть более извилистый участок пути.

В качестве значения векторов в автопилоте может устанавливаться значение скорости, ускорения и силы. В зависимости от системы, в которой установлен автопилот. Например, в корабле также возможен вектор силы течения и ветра.

Действия над векторами

Определение координат вектора

Для определения координат векторов, требуется от координат конечной точки вычесть координаты начальной точки.

$$A(x_1; y_1) \quad B(x_2; y_2) \quad \overline{AB} = (x_2 - x_1; y_2 - y_1)$$

Рис 1.

Умножение вектора на скаляр

Для того чтобы умножить вектор на какое-либо число, необходимо координаты вектора до умножить на это самое число.

$$\vec{a}(x; y) \quad 3 * \vec{a} = (x * 3; y * 3)$$



Рис 2.

Сложение векторов

Для сложения двух векторов, заданных координатами, нужно сложить их соответствующие координаты.

$$\vec{a}(x_1; y_1) \quad \vec{b}(x_2; y_2) \\ \vec{c} = \vec{a} + \vec{b} = (x_1 + x_2; y_1 + y_2)$$

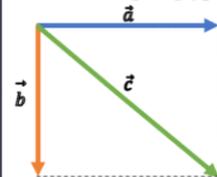


Рис 3.

Вычитание векторов

Для вычитания двух векторов, заданных координатами, нужно вычесть их соответствующие координаты.

$$\vec{a}(x_1; y_1) \quad \vec{b}(x_2; y_2) \\ \vec{c} = \vec{a} - \vec{b} = (x_1 - x_2; y_1 - y_2)$$

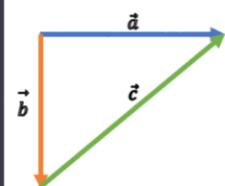


Рис 4.

Задача

Рыболовное судно, находясь на побережье, движется со скоростью, вектор которого равен n . В процессе движения перед ним возникает препятствие в виде кораллов. У судна есть датчики, которое улавливают это препятствие. В конечном итоге судно оплывает кораллы и движется далее. Напишите программный код, в котором будет смоделирована данная ситуация, используйте знания векторной алгебры.

Решение:

Итак, как же можно применить векторную алгебру в написании программного кода для моделирования поведения корабля? Для начала, используем матрицу в качестве пространства для движения корабля. Препятствия в данной матрице пусть будет обозначаться цифрой 1, а пространство, свободное для маневрирования, будет идентифицироваться как 0. Скорость корабля будет определяться как единица пройденного пространства за единицу времени. Так же, введем 3 вектора: a , b , c .

a -вектор движения корабля по диагонали, b - вектор движения вправо, c -вектор движения влево. Для выбора вектора движения будут использованы циклы проверки.

Программный код будет написан на языке C# в среде разработки VisualStudio.

Код программы:

```
using System;

namespace NewProject
{
    class Program
    {
        static void Main(string[] args)
        {
            int nach_x = 0, nach_y = 0;
            int[,] massiv = new int[7,7];
            for (int i = 1; i <= 6; i++)
            {
                for(int j = 1; j <= 6; j++)
                {
                    massiv[i,j] = 0;
                }
            }

            //Расставление препятствий на плоскости
            for (int i = 1; i <= 6; i++)
            {
                for (int j = 1; j <= 6; j++)
                {
                    massiv[3, 3] = 1;
                    massiv[4, 4] = 1;
                    massiv[3, 4] = 1;
                    massiv[4, 2] = 1;
                    massiv[1, 2] = 1;
                }
            }

            Console.WriteLine("Начало старта");
            Console.WriteLine("Точка по координатам x: ");
            nach_x = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Точка по координатам y: ");
            nach_y = Convert.ToInt32(Console.ReadLine());

            Console.WriteLine();

            for (int i = 1; i <= 6; i++)
            {
                for (int j = 1; j <= 6; j++)
                {
                    Console.Write("{0}t", massiv[i, j]);
                }
                Console.WriteLine();
                Console.WriteLine();
            }
            Console.WriteLine();

            for (int i = 1; i <= 6; i++)
            {
                for (int j = 1; j <= 6; j++)
                {
                    if ((massiv[nach_x + 1, nach_y + 1] != 1) && (nach_x + 1 <= 5) && (nach_y + 1 <= 5))
                    {
                        nach_x++;
                        nach_y++;
                    }

                    Console.WriteLine("Координаты вектора A (" + (nach_x + 1) - nach_x + ";" + ((nach_y + 1) - nach_y) + ")");
                    Console.WriteLine("Координаты местоположения (" + nach_x + ";" + nach_y + ")");
                    Console.WriteLine();
                }
                else if ((massiv[nach_x + 1, nach_y] != 1) && (nach_x + 1 <= 5) && (nach_y + 1 <= 5))
                {
                    nach_x++;
                }
                Console.WriteLine("Координаты вектора C (" + (nach_x + 1) - nach_x + ";" + ((nach_y - nach_y) + ")");
                Console.WriteLine("Координаты местоположения (" + nach_x + ";" + nach_y + ")");
                Console.WriteLine();
            }
            else if (((nach_x == 4) && (nach_y == 3)) && ((massiv[nach_x, nach_y + 1] != 1)))
            {
                nach_y++;
            }

            Console.WriteLine("Координаты вектора B (" + (nach_x - nach_x) + ";" + ((nach_y + 1) - nach_y) + ")");
            Console.WriteLine("Координаты местоположения (" + nach_x + ";" + nach_y + ")");
            Console.WriteLine();
        }
        else if ((massiv[nach_x + 1, nach_y + 1] != 1) && (nach_x + 1 <= 5) && (nach_y + 1 <= 5))
        {

```

```

nach_x++;
nach_y++;

Console.WriteLine("Координаты вектора В (" + (nach_x + 1) - nach_x + ";" + ((nach_y + 1) - nach_y) + ")");
Console.WriteLine("Координаты местоположения (" + nach_x + ";" + nach_y + ")");
Console.WriteLine();
}
else if (((nach_x == 5) && (nach_y == 4)) && (nach_x <= 5) && (nach_y + 1 <= 5))
{
nach_y++;

Console.WriteLine("Координаты вектора В (" + (nach_x - nach_x) + ";" + ((nach_y + 1) - nach_y) + ")");
Console.WriteLine("Координаты местоположения (" + nach_x + ";" + nach_y + ")");
Console.WriteLine();
}
}
}
if (((nach_x == 5) && (nach_y == 5)))
{
Console.WriteLine("Поздравляю вы успешно добрались до точки назначения");
}
}
}
}

```

Пояснения:

```

int nach_x = 0, nach_y = 0;
int[,] massiv = new int[7,7];

```

Рис. 5.

Объявление переменных nach_x и nach_y, которые будут использоваться как координаты векторов в программных вычислениях. Далее идет объявление переменной массива massiv и его размерности.

```

for (int i = 1; i <= 6; i++)
{
    for(int j = 1; j <= 6; j++)
    {
        massiv[i,j] = 0;
    }
}

```

Рис 6.

Далее следует заполнение выше объявленного массива massiv нулями. Нулём в программе считается пустое пространство, т.е. пространство без препятствий.

```

for (int i = 1; i <= 6; i++)
{
    for (int j = 1; j <= 6; j++)
    {
        massiv[3, 3] = 1;
        massiv[4, 4] = 1;
        massiv[3, 4] = 1;
        massiv[4, 2] = 1;
        massiv[1, 2] = 1;
    }
}

```

Рис 7.

Расставление препятствий осуществляется через отдельный двумерный цикл, для предотвращения ошибок в ходе выполнения программы. В качестве препятствий используется цифра один.

```

Console.WriteLine("Начало старта");
Console.Write("Точка по координатам x: ");
nach_x = Convert.ToInt32(Console.ReadLine());
Console.Write("Точка по координатам y: ");
nach_y = Convert.ToInt32(Console.ReadLine());

```

Рис 8.

При старте программы пользователю следует ввести начальную точку направления вектора nach_x и nach_y.

```

for (int i = 1; i <= 6; i++)
{
    for (int j = 1; j <= 6; j++)
    {
        Console.WriteLine("{0}\t", massiv[i, j]);
    }
    Console.WriteLine();
    Console.WriteLine();
}

```

Рис 9.

Далее идет отображения массива пользователю.

```

for (int i = 1; i <= 6; i++)
{
    for (int j = 1; j <= 6; j++)
    {
        if ((massiv[nach_x + 1, nach_y + 1] != 1) && (nach_x + 1 <= 5) &&
(nach_y + 1 <= 5))
        {
            nach_x++;
            nach_y++;

            Console.WriteLine("Координаты вектора A (" + ((nach_x + 1) -
nach_x) + ";" + ((nach_y + 1) - nach_y) + ")");
            Console.WriteLine("Координаты местоположения (" + nach_x + "," +
nach_y + ")");
            Console.WriteLine();
        }
        else if ((massiv[nach_x + 1, nach_y] != 1) && (nach_x + 1 <= 5) &&
(nach_y + 1 <= 5))
        {
            nach_x++;

            Console.WriteLine("Координаты вектора C (" + ((nach_x + 1) -
nach_x) + ";" + ((nach_y - nach_y) + ")");
            Console.WriteLine("Координаты местоположения (" + nach_x + "," +
nach_y + ")");
            Console.WriteLine();
        }
        else if (((nach_x == 4) && (nach_y == 3)) && ((massiv[nach_x, nach_y
+ 1] != 1)))
        {
            nach_y++;

            Console.WriteLine("Координаты вектора B (" + (nach_x - nach_x) +
";" + ((nach_y + 1) - nach_y) + ")");
            Console.WriteLine("Координаты местоположения (" + nach_x + "," +
nach_y + ")");
            Console.WriteLine();
        }
        else if ((massiv[nach_x + 1, nach_y + 1] != 1) && (nach_x + 1 <= 5)
&& (nach_y + 1 <= 5))
        {
            nach_x++;
            nach_y++;

            Console.WriteLine("Координаты вектора B (" + ((nach_x + 1) -
nach_x) + ";" + ((nach_y + 1) - nach_y) + ")");
            Console.WriteLine("Координаты местоположения (" + nach_x + "," +
nach_y + ")");
            Console.WriteLine();
        }
        else if (((nach_x == 5) && (nach_y == 4)) && (nach_x <= 5) && (nach_y
+ 1 <= 5))
        {
            nach_y++;

            Console.WriteLine("Координаты вектора B (" + (nach_x - nach_x) +
";" + ((nach_y + 1) - nach_y) + ")");
            Console.WriteLine("Координаты местоположения (" + nach_x + "," +
nach_y + ")");
            Console.WriteLine();
        }
    }
}

```

Рис 10.

Самая главная часть - алгоритм прохождения через существующие препятствия, где проверяется наличие препятствий на пути и их дальнейший обход.

После успешного прохождения цикла проверок выдаться сообщение «Поздравляем, вы успешно добрались до точки назначения».

Данный продукт не является совершенным и представлен как образец возможной реализации процесса автопилотирования корабля путем прохождения его через препятствия такие как: рифы, скалы, корабли, стоящие на месте (допустим в порту) и др. В дальнейшем планируется модернизация и улучшение функционала.

Заключение

Математика в программном обеспечении играет важную роль в постановке и решении задач разного направления. В медицине, навигации, логистике, промышленности, быту и т.п.

Мы рассмотрели принцип работы программного кода с вычислением векторов. И можем сделать вывод, что для внедрения метода решения задачи в программный код требует знания в обеих областях.

Подводя итог, можем сказать, что авторы данной статьи постарались показать, насколько важна и обширна область применения математики в программном обеспечении цифровых систем.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бхаргава А. Грокаем алгоритмы. Иллюстрирование пособие для программистов и любопытствующих//Учебное пособие/ Питер, 2017. С.288.
2. Письменный Д.Т. Конспект лекций по высшей математике: полный курс//Учебное пособие/Москва Айрис пресс, 2013. С 608.
3. Розен Д. Линейная алгебра для разработчиков игр // Электронной ресурс/ Интернет, 2011. С.5
4. Е. Г. Агапова, Н. Б. Лазарева. Векторная алгебра//Учебное пособие/Хабаровск, 2015. С 67.
5. В. М. Безменов Л.Н. Прокофьев. Векторная алгебра //Учебное пособие/Самара, 2001 С 60.

REFERENCES

1. Bhargava A. Grokayem algorithms. Illustrating a guide for programmers and the curious//Training manual/ Piter, 2017. p. 288.
2. Written D. T. Lecture notes on higher mathematics: a complete course//Training manual/Moscow, Iris Press, 2013. P. 608.
3. Rosen D. Linear algebra for game developers // Eelectronnyresurs/ Internet, 2011. p. 5
4. E. G. Agarova, N. B. Lazareva. Vector algebra//Textbook / Khabarovsk, 2015. P. 67.
5. V. M. Bezmenov L. N. Prokofiev. Vector algebra //Training manual/Samara, 2001, P. 60.

Информация об авторах

Боднюк Максим Евгеньевич – студент 1 курса факультета «Управление на транспорте и информационные технологии» специальность «Разработка программно-информационных систем», Иркутский государственный университет путей сообщения, г. Иркутск, e-mail: maks.bodnyuk@mail.ru

Знайдюк Алексей Николаевич - студент 1 курса факультета «Управление на транспорте и информационные технологии» специальность «Разработка программно-информационных систем», Иркутский государственный университет путей сообщения, г. Иркутск, e-mail: Znaidyuk00@gmail.com

Байкова Людмила Анатольевна - старший преподаватель кафедры «Математика», Иркутский государственный университет путей сообщения, г. Иркутск, e-mail: baykova_la@irgups.ru

Authors

Maxim Yevgenyevich Bodnyuk - 1st year student of the Faculty «Transport Management and Information Technologies specialty» Development of «Software and information systems», Irkutsk State University of Railway Transport, Irkutsk, e-mail: maks.bodnyuk@mail.ru

Alexey Nicolayevich Znaydyuk - 1st year student of the Faculty «Transport Management and Information Technologies specialty» Development of «Software and information systems», Irkutsk State University of Railway Transport, Irkutsk, e-mail: Znaidyuk00@gmail.com

Lyudmila Anatolievna Baykova - Senior Lecturer of the Department of «Mathematics», Irkutsk State University of Railway Transport, Irkutsk, e-mail: baykova_la@irgups.ru

Для цитирования

Боднюк М.Е. Применение математики в программировании на примере векторной алгебры [Электронный ресурс] / М. Е. Боднюк, А. Н. Знайдюк, Л. А. Байкова // Молодая наука Сибири: электрон. науч. журн. — 2021. — №12. — Режим доступа: <http://mnv.irgups.ru/toma/121-2021>, свободный. — Загл. с экрана. — Яз. рус., англ. (дата обращения: 07.06.2021)

For citation

Bodnyuk M.Y., Znaydyuk A.N., Baykova L.A. *Application of mathematics in programming using the example of vector algebra. Molodaya nauka Sibiri: ehlektronnyj nauchnyj zhurnal* [Young science of Siberia: electronic scientific journal], 2021, no. 12. [Accessed 07/06/21]