Противовоздушная оборона: математика, информатика, техника

Задача.

Из пушки выпустили снаряд, который нужно перехватить и уничтожить. Для этого через 10 секунд после выстрела пускается ракета, которая летит строго на снаряд и достигает цели. Необходимо, определить траектории полёта снаряда и ракеты.

У нас задача плоская. Но ничто не мешает нам ввести третью координату z и решить трёхмерную задачу (задание продвинутым студентам).

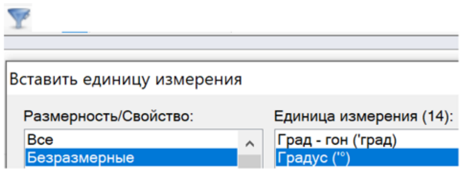
На рисунке 1 показана фотография ночного неба над одним из городов Израиля с траекториями полета ракет системы ПВО «Железный купол».

Рис. 1 – Фото полёта ракет, уничтожающих снаряды

На рисунке 2 первая строка – это свернутая область, где задается функция пользователя, возвращающая плотность воздуха в зависимости от высоты – см. рис. 6 (тема первого занятия).

Далее показан ввод исходных данных для SMath-расчёта полёта ракеты к летящему артиллерийскому снаряду, у которого начальная скоростью v_0. Ствол пушки поднят над горизонтом под углом α. Задана масса снаряда m, диаметр его сечения d (три дюйма) и коэффициент трения снаряда о воздух f (первая строка расчёта). Далее задаются координаты пуска ракеты ПВО к снаряду и её скорость, которая будет оставаться постоянной во время погони за снарядом. Наверху рис. 2 показаны инструменты ввода в расчет единиц измерения. Переменные имеют индексы – текстовые (часть имени переменной, слегка опущенная вниз) и векторные (оператор работы с массивами – векторами и матрицами). Первый индекс вводится в расчёт через точку, а второй – через символ "[" (открывающаяся квадратная скобка – см. иконку справа от константы 15 км). Возможен и двойной индекс – см. операторы, где задаются первые элементы векторов x_r и y_r – координаты старта ракеты ПВО. В ходе расчёта (а он скрыт в свёрнутой области с соответствующей надписью) будут заполнены последующие элементы этих двух векторов, что позволит показать на графике траекторию полёта ракеты к снаряду – см нижнюю часть рис. 2.

⊞— Плотность воздуха по высоте



$v_0 := 700\ \frac{м}{с}$ $\alpha := 60°$ $m := 3\ кг$ $d := 3\ дюйм = 76.2\ мм$ $f := 0.047$

$x_{r_1} := 15\ км$ ⟸ x.r[1] $y_{r_1} := 0\ км$ $v_r := 300\ \frac{м}{с}$

⊞— Здесь записан сам расчёт

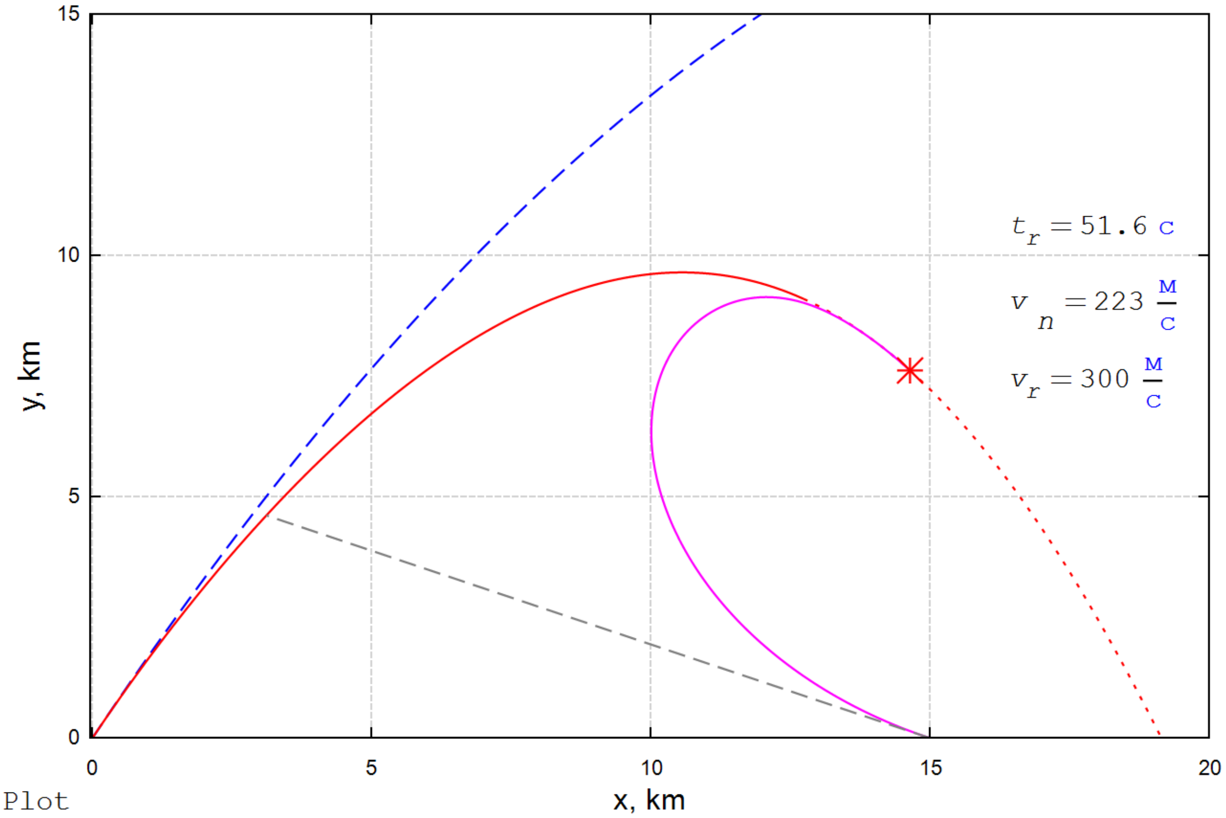


Рис. 2 – Расчёт перехвата артиллерийского снаряда

Если раскрыть свернутую область с именем `Здесь записан сам расчёт`, то мы увидим то, что показано на рис. 3. Отдельные части расчета помещены в отдельные области, что помогает создавать расчёт и комментировать его.

На рисунке 3 записаны две функции, возвращающие координаты полёта снаряда в безвоздушном пространстве в зависимости от времени. Индекс p в именах функций – это первая буква слова «парабола». Снаряд (материальная точка, имеющая массу, но не имеющая размеры) полетит из пушки по параболе, если принять, что земля плоская (однородное гравитационное поле), а воздуха нет. На рисунке 2 показаны две траектории полёта снаряда, выходящие из начала координат: пунктир – без учёта сопротивления воздуха и сплошная линия – с учётом. А как был принят воздух во внимание?

Сначала ведётся расчёт площади поперечного сечения снаряда s и задается примерное время полёта снаряда t_{end}. В переменную N записывается число 1000, нужное для численного решения задачи – для расчета дискретных значений координат снаряда с промежутком времени Δt.

Для этого была составлена и решена система двух обыкновенных дифференциальных уравнений полета снаряда из пушки. Они показаны в на рис. 3 в Mathcad-блоке: произведение массы на ускорение (на вторую производную пути по времени) равно сумме сил, действующих на материальную точку. Задействован принцип суперпозиций: упомянутое равенство (второй закон Ньютона) рассмотрено и по горизонтали, и по вертикали. В горизонтальном направлении на снаряд действует только сила трения воздуха. В вертикальном направлении к этой силе добавлена сила притяжения Земли. Скорость полета снаряда в вертикальном направлении меняет свой знак – снаряд сначала летит вверх, а потом вниз. Поэтому-то в соответствующем дифференциальном уравнении записан не квадрат скорости, а произведение скорости на её абсолютное значение. Это сделано для того, чтобы сила трения снаряда о воздух была всегда направлена против движения снаряда. Возведение во вторую степень лишает скорость знака (направления): вектор превращается в скаляр.

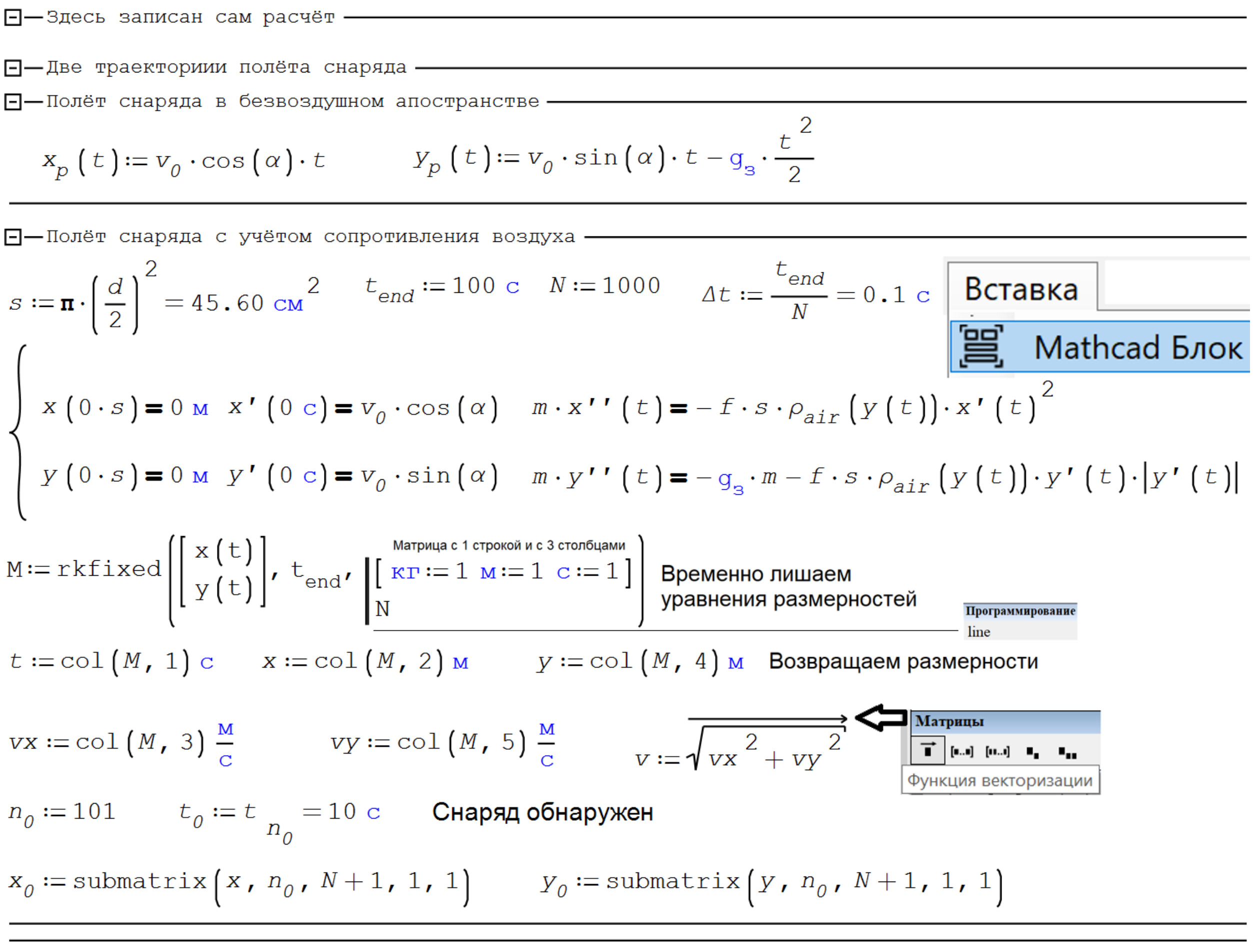


Рис. 3 – Расчёт траектории полета артиллерийского снаряда

Если из уравнений, показанных в рамочке на рис. 3, в правой части убрать произведения, где фигурирует коэффициент f, то такую систему можно решить аналитически – получить функции с индексом p, которые введены в расчёт на рис. 3. Если же учитывать сопротивление воздуха, то аналитического решения можно и не получить. Тут придется переходить к численным методам расчёта – к табулированию искомых функций, что и показано на рис. 3 - используется функция *rkfixed*.

Функция *rkfixed* возвращает вектор с пятью столбцами и N+1 строками. Первый столбец – это дискретные значения времени от нуля (выстрел) до значения t_{end}, которое задано пользователем вместе со значением N. Остальные столбцы матрицы M хранят рассчитанные значения координат летящего снаряда и его скорости по абсциссе и ординате. Встроенная в SMath функция `col` изымает из матрицы указанные столбцы и заносит их в вектора t, x, y, vx и vy. Вектор v хранит полную скорость снаряда.

В расчёте на рисунке 4 формируются векторы x_r и y_r, первые элементы которых (дислокация ракетной установки ПВО) были заданы в начале расчёта на рис. 2.

Переменная Δt хранит промежуток времени, за который снаряд переместиться из одной дискретной точки в другое. Вектор φ будет накапливать значения азимута полета ракеты – угла между направлением её движения и линией горизонта. Переменная L – это расстояние от ракеты до снаряда. Ракета, повторяем, летит строго на снаряд. Она управляется с земли и/или с помощью бортового компьютера.

Векторы x_r, y_r и φ заполняются в теле цикла `for` с четырьмя аргументами (см. иконку на рис.4): мы знаем значения предыдущих элементов трёх векторов (`i`) – рассчитываем по несложным формулам значения последующих элементов (i+1). Такие итерации проводятся до тех пор, пока ракета не достигнет снаряда – расстояние между ними станет меньше пяти метров. Тут сработает боевая часть ракеты и снаряд будет уничтожен – см. разлетающиеся искры на рис. 1: салют в честь сбитого снаряда.

Траектория полёта ракеты ПВО

$$\varphi_1 := \left(\operatorname{arctg}\left(\frac{y_{0_1} - y_{r_1}}{\text{м}}, \frac{x_{0_1} - x_{r_1}}{\text{м}} \right) \right) = [158.8]\ ^{\circ}$$

$$L := \sqrt{\left(x_{0_1} - x_{r_1}\right)^2 + \left(y_{0_1} - y_{r_1}\right)^2} = 12.75\ \text{км}$$

$$\begin{aligned}
&\text{for } i := 1, L > 5\ \text{м}, i := i + 1 \\
&\quad x_{r_{i+1}} := \operatorname{eval}\left(x_{r_i} + v_r \cdot \Delta t \cdot \cos\left(\varphi_i\right) \right) \\
&\quad y_{r_{i+1}} := \operatorname{eval}\left(y_{r_i} + v_r \cdot \Delta t \cdot \sin\left(\varphi_i\right) \right) \\
&\quad \varphi_{i+1} := \operatorname{eval}\left(\operatorname{arctg}\left(\frac{y_{0_i} - y_{r_i}}{\text{м}}, \frac{x_{0_i} - x_{r_i}}{\text{м}} \right) \right) \\
&\quad L := \sqrt{\left(x_{0_{i+1}} - x_{r_{i+1}}\right)^2 + \left(y_{0_{i+1}} - y_{r_{i+1}}\right)^2}
\end{aligned}$$

for

for (3)

for (4)

$$n := \operatorname{length}\left(x_r\right) = 517 \qquad t_r := t_n = 51.6\ \text{с} \qquad L = 1.7609\ \text{м}$$

$$x_1 := \operatorname{submatrix}\left(x, 1, n + n_0, 1, 1\right) \qquad x_2 := \operatorname{submatrix}\left(x, n + n_0, N + 1, 1, 1\right)$$

$$y_1 := \operatorname{submatrix}\left(y, 1, n + n_0, 1, 1\right) \qquad y_2 := \operatorname{submatrix}\left(y, n + n_0, N + 1, 1, 1\right)$$

Рис. 4 – Расчёт траектории полета ракеты ПВО

Особо следует рассказать о функции `eval` в среде SMath. Дело в том, что в этот физико-математический пакет встроена символьная, а не численная математика. Вследствие этого вычисления производятся с максимальной точностью и, следовательно, довольно долго. Это незаметно при линейных вычислениях, но становится нестерпимым в циклах. Функция `eval` преобразует символьное выражение (1/3, например) в численное (0.333…), что ускоряет расчёт за счёт некоторой потери точности. Но у нас весь расчёт довольно приближенный даже без учета принятых упрощений.

Последний оператор области, показанной в свернутом виде на рис. 2, отображен на рис. 5. Переменную *Plot* достаточно вставить аргументом графика и отформатировать его должным образом, чтобы получить кривые, показанные не рис. 2.

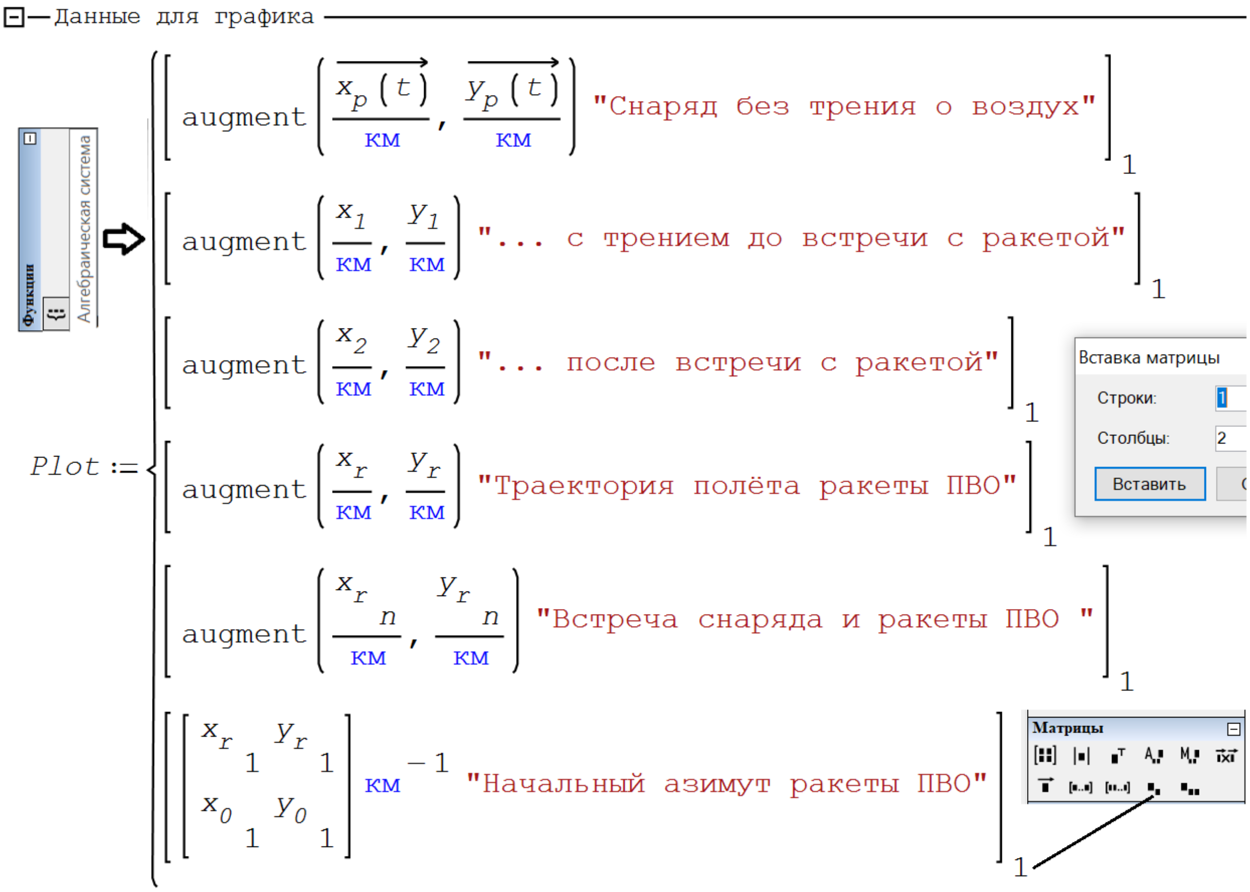
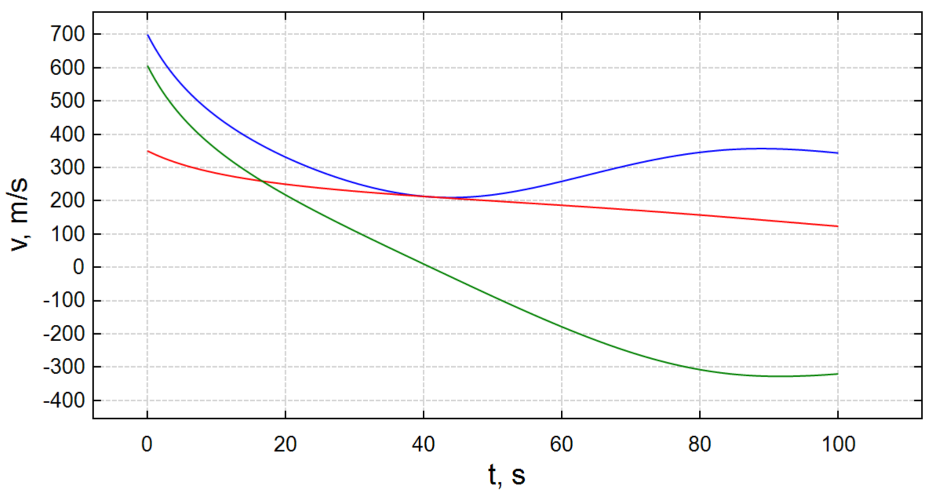


Рис. 5 – Формирование аргумента графика на рис. 2

В конце расчета можно показать, как меняется скорость снаряда, сбиваемого ракетой ПВО. Синяя кривая – это сама скорость, а красная и зелёная кривые – это проекции скорости по горизонтали и вертикали



$$\begin{cases} \mathrm{augment}(t, v) \\ \mathrm{augment}(t, vx) \\ \mathrm{augment}(t, vy) \end{cases}$$

Примечание.

На рисунке 3 были показаны две функции с именами x_p и y_p, по которым строился параметрический график полета снаряда без учёта сопротивления воздуха – см. верхнюю пунктирную кривую на рис. 2. На рисунке 6 показано, как были получены эти функции через аналитическое решение системы двух дифференциальных уравнений полёта снаряда без учёта сопротивления воздуха, когда на снаряд действует только одна сила – сила притяжения в вертикальном направлении. Используется функция *dsolve*, предназначенная для этих целей. Она подсоединяется к пакету SMath Studio через приложение (плагин) maple. Расчет на рис. 6 нужно выполнять либо на отдельном листе расчёта, либо в самом начале расчета , описанного выше.

$$\mathrm{maple}\left(\mathit{dsolve}\left(\left(\begin{cases} m\cdot\frac{d^2}{dt^2}x(t)=0\\ m\cdot\frac{d^2}{dt^2}y(t)=-m\cdot g_3\\ x(0)=0\\ D(x)(0)=v_0\cdot\cos(\alpha)\\ y(0)=0\\ D(y)(0)=v_0\cdot\sin(\alpha)\end{cases}\right)\right)\right)=\begin{bmatrix} v_0\cdot\cos(\alpha)\cdot t\\ \frac{t\cdot\left(-g_3\cdot t+2\cdot v_0\cdot\sin(\alpha)\right)}{2}\end{bmatrix}$$

D [Ctrl+Shift+k] (x) (0) [Ctrl+Shift+k]

D [Ctrl+Shift+k] (y) (0) [Ctrl+Shift+k]

Рис. 6 – Аналитическое решение системы дифференциальных уравнений полёта снаряда в безвоздушном пространстве