**Приложение А**

Ниже представлены результаты интегрирования для траектории без коррекции.

Таблица А.1 – Параметры номинальной траектории полета мины

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| t,  s | x,  m | y,  m | z,  m | Vx,  m/s | Vy,  m/s | Vz,  m/s | V,  m/s | ϑ,  grad | ψ,  grad | γ,  grad | ωx,  grad/s | ωy,  grad/s | ωz,  grad/s | α,  grad | β,  grad |
| 0.00 | 0.00 | 0.00 | 0.00 | 152.507 | 419.009 | 0.000 | 445.900 | 70.000 | 0.000 | 0.000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 5.00 | 631.34 | 1623.68 | -0.02 | 110.325 | 259.730 | -0.005 | 282.191 | 67.069 | -0.022 | -61.453 | -244.0840 | 0.6771 | -0.3683 | 0.0313 | -0.0780 |
| 10.00 | 1148.71 | 2723.61 | -0.05 | 98.012 | 184.181 | -0.008 | 208.636 | 62.170 | -0.051 | -33.711 | -181.0560 | 0.6825 | -1.0266 | 0.1435 | -0.1269 |
| 15.00 | 1619.78 | 3489.09 | -0.10 | 90.981 | 123.530 | -0.013 | 153.418 | 54.180 | -0.139 | -97.609 | -134.4450 | 2.0521 | 0.2643 | -0.1593 | -0.5356 |
| 20.00 | 2062.74 | 3969.79 | -0.18 | 86.519 | 69.500 | -0.020 | 110.977 | 40.214 | -0.268 | 45.105 | -98.3204 | -2.6234 | -2.5796 | 1.1713 | 0.8647 |
| 25.00 | 2487.32 | 4189.55 | -0.30 | 83.461 | 18.766 | -0.030 | 85.545 | 15.754 | -0.487 | -21.946 | -75.1520 | 2.2726 | -5.7472 | 2.6734 | -1.6097 |
| 30.00 | 2898.06 | 4160.43 | -0.48 | 80.848 | -30.157 | -0.043 | 86.290 | -18.356 | -0.572 | -21.709 | -72.8334 | 2.4817 | -6.1178 | 1.7432 | -1.3013 |
| 35.00 | 3295.32 | 3891.05 | -0.72 | 77.944 | -77.230 | -0.053 | 109.726 | -43.803 | -0.267 | -66.419 | -91.7058 | 3.4714 | -1.5012 | 0.1745 | -0.9428 |
| 40.00 | 3676.06 | 3393.09 | -1.00 | 74.189 | -121.346 | -0.058 | 142.228 | -58.122 | -0.132 | 126.246 | -119.9230 | -1.7203 | 1.2570 | -0.1844 | 0.4076 |
| 45.00 | 4035.52 | 2685.09 | -1.30 | 69.424 | -160.980 | -0.060 | 175.312 | -66.455 | -0.055 | 173.485 | -149.0120 | -0.1592 | 1.2964 | -0.2108 | 0.0659 |
| 50.00 | 4368.60 | 1793.52 | -1.60 | 63.645 | -194.514 | -0.059 | 204.662 | -71.764 | -0.003 | 81.485 | -174.9650 | -0.8562 | -0.1311 | 0.0348 | 0.1144 |
| 58.32 | 4851.36 | 0.00 | -2.06 | 51.931 | -232.136 | -0.051 | 237.873 | -77.334 | 0.036 | 1.888 | -182.4520 | 0.0122 | -0.5310 | 0.0562 | -0.0026 |

На рисунках А.1-А.11 представлены графические зависимости полученной траектории.

Рисунок А.1 – Траектория в проекции на вертикальную плоскость ОХУ

Рисунок А.2 – Траектория в проекции на горизонтальную плоскость ОХZ

Рисунок А.3 – Зависимость угла атаки от времени полета

Рисунок А.4 – Зависимость угла скольжения от времени полета

Рисунок А.5 – Зависимость угла тангажа от времени полета

Рисунок А.6 – Зависимость угла рысканья от времени полета

Рисунок А.7 – Зависимость угла крена от времени полета

Рисунок А.8 – Зависимость угловой скорости по оси Х от времени полета

Рисунок А.9 – Зависимость угловой скорости по оси Y от времени полета

Рисунок А.10 – Зависимость угловой скорости по оси Z от времени полета

Рисунок А.11 – Плоскость углов скольжения-атаки

Ниже представлены графические зависимости параметров траектории для «зоны коррекции». Сброс головного обтекателя происходит при времени полета «Зона коррекции» начинается при времени полета 55.3 секунды.

Рисунок Б.1 – Траектория в проекции на вертикальную плоскость ОХУ

Рисунок Б.2 – Траектория в проекции на горизонтальную плоскость ОХZ

Рисунок Б.3 – Зависимость угла атаки от времени полета

Рисунок Б.4 – Зависимость угла скольжения от времени полета

Рисунок Б.5 – Зависимость угла тангажа от времени полета

Рисунок Б.6 – Зависимость угла рысканья от времени полета

Рисунок Б.7 – Зависимость угла крена от времени полета

Рисунок Б.8 – Зависимость угловой скорости по оси Х от времени полета

Рисунок Б.9 – Зависимость угловой скорости по оси Y от времени полета

Рисунок Б.10 – Зависимость угловой скорости по оси Z от времени полета

Рисунок Б.11 – Плоскость углов скольжения-атаки

Рисунок Б.12 – Зависимость угла Пеленга от времени полета

Рисунок Б.13 – Зависимость расстояния до цели от времени полета

**Приложение Б**

Ниже представлены результаты интегрирования для траектории после сброса головного обтекателя. Координаты цели приняты х = 4865 м, z =  -7 м.

Таблица Б.1 – Параметры траектории полета мины после сброса обтекателя

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| t,  s | x,  m | y,  m | z,  m | Vx,  m/s | Vy,  m/s | Vz,  m/s | V,  m/s | ϑ,  grad | ψ,  grad | γ,  grad | ωx,  grad/s | ωy,  grad/s | ωz,  grad/s | α,  grad | β,  grad |
| 54.92 | 4665.95 | 770.09 | -1.88 | 57.097 | -220.193 | -0.055 | 227.475 | -75.386 | 0.017 | -109.565 | -182.2130 | 0.5896 | 0.2185 | -0.0348 | -0.0697 |
| 55.12 | 4677.34 | 725.97 | -1.89 | 56.797 | -221.000 | -0.054 | 228.181 | -75.513 | 0.033 | -145.275 | -176.6130 | 0.3430 | 0.5455 | -0.0641 | -0.0379 |
| 55.32 | 4688.67 | 681.69 | -1.90 | 56.497 | -221.793 | -0.053 | 228.876 | -75.644 | 0.055 | 179.525 | -175.9120 | -0.0260 | 0.6677 | -0.0650 | 0.0002 |
| 55.52 | 4699.94 | 637.25 | -1.91 | 56.197 | -222.573 | -0.053 | 229.558 | -75.777 | 0.056 | 144.325 | -176.1650 | -0.3682 | 0.5455 | -0.0428 | 0.0300 |
| 55.72 | 4711.15 | 592.66 | -1.92 | 55.896 | -223.341 | -0.053 | 230.229 | -75.904 | 0.028 | 109.021 | -176.6040 | -0.5552 | 0.2392 | -0.0085 | 0.0449 |
| 55.92 | 4722.30 | 547.92 | -1.93 | 55.594 | -224.095 | -0.053 | 230.888 | -76.018 | -0.005 | 73.621 | -177.0960 | -0.5307 | -0.1293 | 0.0276 | 0.0430 |
| 56.12 | 4733.39 | 503.02 | -1.94 | 55.291 | -224.837 | -0.053 | 231.535 | -76.126 | -0.008 | 38.147 | -177.6080 | -0.3159 | -0.4321 | 0.0554 | 0.0246 |
| 56.32 | 4744.41 | 457.98 | -1.95 | 54.988 | -225.565 | -0.052 | 232.171 | -76.236 | 0.018 | 2.600 | -178.1120 | 0.0100 | -0.5700 | 0.0646 | -0.0057 |
| 56.52 | 4755.38 | 412.80 | -1.96 | 54.685 | -226.280 | -0.053 | 232.794 | -76.353 | 0.039 | -33.050 | -178.5900 | 0.3330 | -0.4950 | 0.0494 | -0.0366 |
| 56.72 | 4766.29 | 367.47 | -1.97 | 54.380 | -226.982 | -0.053 | 233.406 | -76.471 | 0.032 | -68.821 | -179.0450 | 0.5340 | -0.2307 | 0.0150 | -0.0543 |
| 56.92 | 4777.13 | 322.01 | -1.98 | 54.075 | -227.672 | -0.053 | 234.005 | -76.584 | 0.013 | -104.693 | -179.4960 | 0.5334 | 0.1242 | -0.0237 | -0.0510 |
| 57.12 | 4787.92 | 276.40 | -1.99 | 53.769 | -228.348 | -0.052 | 234.593 | -76.692 | 0.013 | -140.637 | -179.9490 | 0.3302 | 0.4294 | -0.0509 | -0.0292 |
| 57.32 | 4800.34 | 231.07 | -2.88 | 71.300 | -224.754 | -9.295 | 235.976 | -76.434 | 0.610 | -176.545 | -185.7330 | -2.2824 | -4.6627 | 3.9252 | 2.3136 |
| 57.52 | 4814.56 | 186.06 | -4.73 | 70.920 | -225.395 | -9.285 | 236.472 | -74.588 | 3.679 | 148.914 | -188.0980 | 1.8848 | -13.3603 | 2.4376 | -0.1352 |
| 57.72 | 4828.71 | 140.91 | -6.59 | 70.521 | -226.045 | -9.236 | 236.970 | -72.052 | 7.580 | 115.315 | -184.5190 | 8.2342 | -9.7176 | -0.2385 | 0.4212 |
| 57.92 | 4842.77 | 95.64 | -8.43 | 70.059 | -226.702 | -9.171 | 237.457 | -70.318 | 10.091 | 80.651 | -186.7580 | 5.6178 | -1.7620 | -0.3853 | 2.4843 |
| 58.12 | 4854.99 | 49.53 | -8.84 | 54.490 | -233.484 | 3.517 | 239.784 | -70.432 | 10.253 | 42.816 | -193.8290 | -7.0458 | -3.9302 | 2.8393 | 6.9139 |
| 58.32 | 4865.84 | 2.77 | -8.12 | 54.061 | -234.041 | 3.660 | 240.232 | -73.320 | 6.735 | 0.370 | -195.3680 | -8.5214 | -19.5103 | 3.8442 | 2.4046 |
| 58.33 | 4866.49 | -0.01 | -8.08 | 54.039 | -234.073 | 3.664 | 240.258 | -73.553 | 6.380 | -2.275 | -195.1680 | -7.9529 | -20.2380 | 3.7078 | 2.1558 |

Продолжение таблицы Б.1

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| t,  s | ξ,  grad | χ,  grad | r,  m | Двигатель сработал | | | | | | | |
| №1 | №2 | №3 | №4 | №5 | №6 | №7 | №8 |
| 54.92 | 0.384 | 1.099 | 795.41 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 55.12 | 0.382 | 56.207 | 749.85 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 55.32 | 0.427 | 110.588 | 704.14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 55.52 | 0.523 | 160.589 | 658.30 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 55.72 | 0.654 | -154.752 | 612.33 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 55.92 | 0.803 | -113.454 | 566.22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 56.12 | 0.966 | -73.522 | 519.98 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 56.32 | 1.158 | -34.163 | 473.62 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 56.52 | 1.394 | 4.521 | 427.14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 56.72 | 1.683 | 42.409 | 380.53 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 56.92 | 2.036 | 79.778 | 333.82 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 57.12 | 2.483 | 116.987 | 287.00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 57.32 | 2.233 | 154.979 | 239.98 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 57.52 | 0.372 | -20.136 | 192.79 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 57.72 | 3.996 | 39.111 | 145.51 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 57.92 | 7.590 | 75.222 | 98.20 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 58.12 | 9.587 | 111.935 | 50.56 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 58.32 | 40.058 | 149.194 | 3.11 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 58.33 | 103.876 | 151.662 | 1.83 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |

**Приложение В**

Ниже представлен листинг программы. Программа написана в среде Microsoft Visual Studio 2017 на языке С++. Листинг модуля атмосферы, расчет матриц перехода и интерполяция аэродинамических коэффициентов опущены.

#include "stdafx.h"

#include <iostream>

#include <cmath>

#include <iomanip>

#include <fstream>

#include <vector>

#define \_USE\_MATH\_DEFINES

#include "math.h"

#include "GOST4401-81.h"

using namespace std;

ofstream txtfile2("2.txt");

ofstream txtfile("1.txt");

#define dtdt K[0]

#define dxgdt K[1]

#define dygdt K[2]

#define dzgdt K[3]

#define dVxgdt K[4]

#define dVygdt K[5]

#define dVzgdt K[6]

#define domegaxdt K[7]

#define domegaydt K[8]

#define domegazdt K[9]

#define droRGdt K[10]

#define dlamdaRGdt K[11]

#define dmuRGdt K[12]

#define dnuRGdt K[13]

#define t par[0]

#define xg par[1]

#define yg par[2]

#define zg par[3]

#define Vxg par[4]

#define Vyg par[5]

#define Vzg par[6]

#define omegax par[7]

#define omegay par[8]

#define omegaz par[9]

#define roRG par[10]

#define lamdaRG par[11]

#define muRG par[12]

#define nuRG par[13]

#define t\_RK parbuf[0]

#define xg\_RK parbuf[1]

#define yg\_RK parbuf[2]

#define zg\_RK parbuf[3]

#define Vxg\_RK parbuf[4]

#define Vyg\_RK parbuf[5]

#define Vzg\_RK parbuf[6]

#define omegax\_RK parbuf[7]

#define omegay\_RK parbuf[8]

#define omegaz\_RK parbuf[9]

#define roRG\_RK parbuf[10]

#define lamdaRG\_RK parbuf[11]

#define muRG\_RK parbuf[12]

#define nuRG\_RK parbuf[13]

// Объявление функций

double linInterp(double x1, double x2, double y1, double y2, double x);

double Cx(double M, double alpha);

double Cy\_alpha(double M, double alpha);

double Cz\_betta(double M, double betta);

double mx\_omegax(double M, double alpha);

double mz\_omegaz(double M, double alpha);

double mz\_alpha(double M, double alpha);

double my\_omegay(double M, double betta);

double my\_betta(double M, double betta);

double mx(double M, double alpha);

double Cx\_update(double M, double alpha);

double Cy\_alpha\_update(double M, double alpha);

double Cz\_betta\_update(double M, double betta);

double mx\_omegax\_update(double M, double alpha);

double mz\_omegaz\_update(double M, double alpha);

double mz\_alpha\_update(double M, double alpha);

double my\_omegay\_update(double M, double betta);

double my\_betta\_update(double M, double betta);

double mx\_update(double M, double alpha);

void MATR(double ro, double lamda, double mu, double nu, double(\*A)[3][3], double(\*AT)[3][3]);

double fromPItoPI(double angle);

const double g = 9.80665;

double K[99], par[99], parbuf[99], par\_drob[99],

A[3][3] = { 0 }, AT[3][3] = { 0 }, B[3][3] = { 0 };

double tang, risk, kren, m, V0, q, M, r, V, X, Y, Z, Mx, My, Mz, Mstab, alpha, betta, alpha\_full, vx, vy, vz, vr, vfi, vhi,

delta\_tang, delta\_risk, delta\_kren, x\_t, y\_t, z\_t, Vx\_t = 0, Vy\_t = 0, Vz\_t = 0, Ksi, hi, x, y, z, rg, ky = 0, kz = 0, t\_on=1E10,

Ix = 0.041,

Iy = 0.753,

Iz = 0.753,

dm = 0.12,

Sm = M\_PI\*dm\*dm / 4.0,

L = 0.90,

Kvr = 0.1,

dhi = 0.03\*M\_PI / 180,

t\_nominal = 58.3,

t\_GO = t\_nominal - 3.5;

bool ENGINE\_WORKS = false;

int i\_engine;

int main()

{

Atm\_GOST();

//Начальные условия

double N = 3,

tang = 70 \* M\_PI / 180.0;

risk = 0;

kren = 0;

m = 20;

V0 = 445 + 0.3\*N;

double colOfEngine = 8;

double deltahi = 2 \* M\_PI / 8;

double hi\_engine[8]; bool engine\_done[8] = { false };

for (int i = 0; i <= 7; i++) hi\_engine[i] = i\*deltahi;

double P\_ud = 500;

double t\_engine = 0.04;

double Ksi\_min = 2.5\*M\_PI / 180;

double Ksi\_max = 13 \* M\_PI / 180;

double dt = 0.001;

// номинал

// x\_t = 4851 z = -2

x\_t = 4865;

y\_t = 0;

z\_t = -7;

t = 0;

xg = 0;

yg = 1e-3;

zg = 0;

Vxg = V0\*cos(tang);

Vyg = V0\*sin(tang);

Vzg = 0;

omegax = 0;

omegay = 0;

omegaz = 0;

roRG = cos(risk\*0.5)\*cos(tang\*0.5)\*cos(kren\*0.5) - sin(risk\*0.5)\*sin(kren\*0.5)\*sin(tang\*0.5);

lamdaRG = sin(risk\*0.5)\*sin(tang\*0.5)\*cos(kren\*0.5) + cos(risk\*0.5)\*cos(tang\*0.5)\*sin(kren\*0.5);

muRG = sin(risk\*0.5)\*cos(tang\*0.5)\*cos(kren\*0.5) + cos(risk\*0.5)\*sin(tang\*0.5)\*sin(kren\*0.5);

nuRG = cos(risk\*0.5)\*sin(tang\*0.5)\*cos(kren\*0.5) - sin(risk\*0.5)\*cos(tang\*0.5)\*sin(kren\*0.5);

txtfile << "t,s" << ";" << "x,m" << ";" << "y,m" << ";" << "z,m" << ";" << "Vx,m/s" << ";" << "Vy,m/s" << ";" << "Vz,m/s" << ";" << "V,m/s" << ";"

<< "tang,grad" << ";" << "risk,grad" << ";" << "kren,grad" << ";" << "omega\_x,grad/s" << ";" << "omega\_y,grad/s" << ";" << "omega\_z,grad/s" << ";"

<< "alpha,grad" << ";" << "betta,grad" << ";" << "Ksi, grad" << ";" << "hi, grad" << ";" << "rg, m" << ";" << "ENGINE\_WORKS" << ";" << "engine\_done" << endl;

// Интегрирование траектории до сброса ГО

int step = 0;

for (;;) {

// Выход из цикла

if (abs(t - t\_GO) < 1e-5) break;

// Дробление шага

if (t - t\_GO > 0) {

dt \*= 0.5;

for (int i = 0; i <= 13; i++) par[i] = par\_drob[i];

}

else {

for (int i = 0; i <= 13; i++) par\_drob[i] = par[i];

// Расчет промежуточных параметров на шаге

MATR(roRG, lamdaRG, muRG, nuRG, &A, &AT);

tang = asin(2 \* (roRG\*nuRG + lamdaRG\*muRG));

risk = atan2(2 \* (roRG\*muRG - lamdaRG\*nuRG), roRG\*roRG + lamdaRG\*lamdaRG - nuRG\*nuRG - muRG\*muRG);

kren = atan2(2 \* (roRG\*lamdaRG - muRG\*nuRG), roRG\*roRG - lamdaRG\*lamdaRG - nuRG\*nuRG + muRG\*muRG);

r = pow((x\_t - xg)\*(x\_t - xg) + (y\_t - yg)\*(y\_t - yg) + (z\_t - zg)\*(z\_t - zg), 0.5),

V = pow(Vxg\*Vxg + Vyg\*Vyg + Vzg\*Vzg, 0.5),

q = fro(yg)\*V\*V\*0.5,

vx = Vxg\*AT[0][0] + Vyg\*AT[0][1] + Vzg\*AT[0][2],

vy = Vxg\*AT[1][0] + Vyg\*AT[1][1] + Vzg\*AT[1][2],

vz = Vxg\*AT[2][0] + Vyg\*AT[2][1] + Vzg\*AT[2][2],

alpha = -atan2(vy, vx);

betta = asin(vz / V);

alpha\_full = sqrt(alpha\*alpha + betta\*betta),

M = V / fa(yg);

X = -Cx(M, alpha\_full)\*q\*Sm;

Y = (Cy\_alpha(M, alpha)\*alpha)\*q\*Sm;

Z = (Cz\_betta(M, betta)\*betta)\*q\*Sm;

Mx = (mx\_omegax(M, alpha\_full)\*omegax\*dm / V + Kvr\*mx(M, alpha\_full))\*q\*Sm\*dm;

My = (my\_omegay(M, betta)\*omegay\*dm / V + my\_betta(M, betta)\*betta)\*q\*Sm\*L;

Mz = (mz\_omegaz(M, alpha)\*omegaz\*dm / V + mz\_alpha(M, alpha)\*alpha)\*q\*Sm\*L;

cout << yg << endl;

//вывод в файл

int st = 5 / dt;

if (step % st == 0)

txtfile

<< t << ";" << xg << ";" << yg << ";" << zg << ";" << Vxg << ";" << Vyg << ";" << Vzg << ";" << V << ";"

<< tang \* 180 / M\_PI << ";" << risk \* 180 / M\_PI << ";" << kren \* 180 / M\_PI << ";"

<< omegax\*180/M\_PI << ";" << omegay \* 180 / M\_PI << ";" << omegaz \* 180 / M\_PI << ";"

<< alpha \*180/M\_PI << ";" << betta \* 180 / M\_PI << ";"

<< endl;

}

step++;

//Производные

dtdt = 1 \* dt,

dxgdt = Vxg\*dt,

dygdt = Vyg\*dt,

dzgdt = Vzg\*dt,

dVxgdt = 1 / m\*(X\*A[0][0] + Y\*A[0][1] + Z\*A[0][2])\*dt,

dVygdt = 1 / m\*(X\*A[1][0] + Y\*A[1][1] + Z\*A[1][2] - m\*g)\*dt,

dVzgdt = 1 / m\*(X\*A[2][0] + Y\*A[2][1] + Z\*A[2][2])\*dt,

domegaxdt = (Mx / Ix - omegay\*omegaz\*(Iz - Iy) / Ix)\*dt,

domegaydt = (My / Iy - omegax\*omegaz\*(Ix - Iz) / Iy)\*dt,

domegazdt = (Mz / Iz - omegay\*omegax\*(Iy - Ix) / Iz)\*dt,

droRGdt = -(omegax\*lamdaRG + omegay\*muRG + omegaz\*nuRG)\*0.5\*dt,

dlamdaRGdt = (omegax\*roRG - omegay\*nuRG + omegaz\*muRG)\*0.5\*dt,

dmuRGdt = (omegax\*nuRG + omegay\*roRG - omegaz\*lamdaRG)\*0.5\*dt,

dnuRGdt = (-omegax\*muRG + omegay\*lamdaRG + omegaz\*roRG)\*0.5\*dt;

// Приращения

for (int i = 0; i <= 13; i++) par[i] += K[i];

// Нормировка параметров РГ

double norm\_RG = pow(roRG\*roRG + lamdaRG\*lamdaRG + nuRG\*nuRG + muRG\*muRG, 0.5);

roRG /= norm\_RG;

lamdaRG /= norm\_RG;

nuRG /= norm\_RG;

muRG /= norm\_RG;

}

// Конец интегрирования траектории до сброса ГО

m = 19;

L = 0.78;

dt = 0.0001;

// ПОСЛЕ СБРОСА ГО

for (;;) {

// Выход из цикла

if (yg < 1E-5) break;

// Дробление шага

if (yg < 0) {

dt \*= 0.5;

for (int i = 0; i <= 13; i++) par[i] = par\_drob[i];

}

else {

for (int i = 0; i <= 13; i++) par\_drob[i] = par[i];

//Выключение двигателя

if ((ENGINE\_WORKS == true) && (t >= t\_on + t\_engine)) {

ENGINE\_WORKS = false;

ky = 0;

kz = 0;

t\_on = 1E10;

}

//Пересчет k1 k2

if (ENGINE\_WORKS == true) {

ky = cos(hi\_engine[i\_engine] + kren);

kz = sin(hi\_engine[i\_engine] + kren);

}

// Расчет промежуточных параметров на шаге

MATR(roRG, lamdaRG, muRG, nuRG, &A, &AT);

tang = asin(2 \* (roRG\*nuRG + lamdaRG\*muRG));

risk = atan2(2 \* (roRG\*muRG - lamdaRG\*nuRG), roRG\*roRG + lamdaRG\*lamdaRG - nuRG\*nuRG - muRG\*muRG);

kren = atan2(2 \* (roRG\*lamdaRG - muRG\*nuRG), roRG\*roRG - lamdaRG\*lamdaRG - nuRG\*nuRG + muRG\*muRG);

V = pow(Vxg\*Vxg + Vyg\*Vyg + Vzg\*Vzg, 0.5),

q = fro(yg)\*V\*V\*0.5,

vx = Vxg\*AT[0][0] + Vyg\*AT[0][1] + Vzg\*AT[0][2],

vy = Vxg\*AT[1][0] + Vyg\*AT[1][1] + Vzg\*AT[1][2],

vz = Vxg\*AT[2][0] + Vyg\*AT[2][1] + Vzg\*AT[2][2],

alpha = -atan2(vy, vx);

betta = asin(vz / V);

alpha\_full = sqrt(alpha\*alpha + betta\*betta);

M = V / fa(yg);

X = -Cx\_update(M, alpha\_full)\*q\*Sm;

Y = (Cy\_alpha\_update(M, alpha)\*alpha)\*q\*Sm + ky\*P\_ud\*g;

Z = (Cz\_betta\_update(M, betta)\*betta)\*q\*Sm + kz\*P\_ud\*g;

Mx = (mx\_omegax\_update(M, alpha\_full)\*omegax\*dm / V + Kvr\*mx\_update(M, alpha\_full))\*q\*Sm\*dm;

My = (my\_omegay\_update(M, betta)\*omegay\*dm / V + my\_betta\_update(M, betta)\*betta)\*q\*Sm\*L;

Mz = (mz\_omegaz\_update(M, alpha)\*omegaz\*dm / V + mz\_alpha\_update(M, alpha)\*alpha)\*q\*Sm\*L;

cout << yg << endl;

// КОРРЕКТИРОВКА\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

rg = pow((x\_t - xg)\*(x\_t - xg) + (y\_t - yg)\*(y\_t - yg) + (z\_t - zg)\*(z\_t - zg), 0.5);

x = (x\_t - xg)\*AT[0][0] + (y\_t - yg)\*AT[0][1] + (z\_t - zg)\*AT[0][2];

y = (x\_t - xg)\*AT[1][0] + (y\_t - yg)\*AT[1][1] + (z\_t - zg)\*AT[1][2];

z = (x\_t - xg)\*AT[2][0] + (y\_t - yg)\*AT[2][1] + (z\_t - zg)\*AT[2][2];

r = sqrt(y\*y + z\*z);

Ksi = atan2(r, x);

Ksi = fromPItoPI(Ksi);

hi = atan2(z, y);

hi = fromPItoPI(hi);

**if (rg > 50)**

**if ((ENGINE\_WORKS == false) && (t >= t\_nominal - 3))**

**if ((Ksi < Ksi\_max) && (Ksi > Ksi\_min))**

**for (int i = 0; i <= 7; i++)**

**if ((abs(fromPItoPI(hi - omegax\*t\_engine\*0.5) - fromPItoPI(hi\_engine[i] + kren)) <= dhi) && (engine\_done[i] == false)) {**

**ENGINE\_WORKS = true;**

**engine\_done[i] = true;**

**i\_engine = i;**

**ky = cos(hi\_engine[i\_engine] + kren);**

**kz = sin(hi\_engine[i\_engine] + kren);**

**t\_on = t;**

**break;**

**}**

//вывод в файл

int st = 0.2 / dt;

if (step % st == 0) {

txtfile

<< t << ";" << xg << ";" << yg << ";" << zg << ";" << Vxg << ";" << Vyg << ";" << Vzg << ";" << V << ";"

<< tang \* 180 / M\_PI << ";" << risk \* 180 / M\_PI << ";" << kren \* 180 / M\_PI << ";"

<< omegax \* 180 / M\_PI << ";" << omegay \* 180 / M\_PI << ";" << omegaz \* 180 / M\_PI << ";"

<< alpha \* 180 / M\_PI << ";" << betta \* 180 / M\_PI << ";"

<< Ksi \* 180 / M\_PI << ";" << hi \* 180 / M\_PI << ";" << rg << ";" << ENGINE\_WORKS << ";";

for (int i = 0; i < 8; i++) txtfile << engine\_done[i] << ";";

txtfile << endl;

}

}

step++;

//Производные

dtdt = 1 \* dt,

dxgdt = Vxg\*dt,

dygdt = Vyg\*dt,

dzgdt = Vzg\*dt,

dVxgdt = 1 / m\*(X\*A[0][0] + Y\*A[0][1] + Z\*A[0][2])\*dt,

dVygdt = 1 / m\*(X\*A[1][0] + Y\*A[1][1] + Z\*A[1][2] - m\*g)\*dt,

dVzgdt = 1 / m\*(X\*A[2][0] + Y\*A[2][1] + Z\*A[2][2])\*dt,

domegaxdt = (Mx / Ix - omegay\*omegaz\*(Iz - Iy) / Ix)\*dt,

domegaydt = (My / Iy - omegax\*omegaz\*(Ix - Iz) / Iy)\*dt,

domegazdt = (Mz / Iz - omegay\*omegax\*(Iy - Ix) / Iz)\*dt,

droRGdt = -(omegax\*lamdaRG + omegay\*muRG + omegaz\*nuRG)\*0.5\*dt,

dlamdaRGdt = (omegax\*roRG - omegay\*nuRG + omegaz\*muRG)\*0.5\*dt,

dmuRGdt = (omegax\*nuRG + omegay\*roRG - omegaz\*lamdaRG)\*0.5\*dt,

dnuRGdt = (-omegax\*muRG + omegay\*lamdaRG + omegaz\*roRG)\*0.5\*dt;

// Приращения

for (int i = 0; i <= 13; i++) par[i] += K[i];

// Нормировка параметров РГ

double norm\_RG = pow(roRG\*roRG + lamdaRG\*lamdaRG + nuRG\*nuRG + muRG\*muRG, 0.5);

roRG /= norm\_RG;

lamdaRG /= norm\_RG;

nuRG /= norm\_RG;

muRG /= norm\_RG;

}

// КОНЕЦ ИНТЕГРИРОВАНИЯ

// Вывод последнего шага

txtfile

<< t << ";" << xg << ";" << yg << ";" << zg << ";" << Vxg << ";" << Vyg << ";" << Vzg << ";" << V << ";"

<< tang \* 180 / M\_PI << ";" << risk \* 180 / M\_PI << ";" << kren \* 180 / M\_PI << ";"

<< omegax \* 180 / M\_PI << ";" << omegay \* 180 / M\_PI << ";" << omegaz \* 180 / M\_PI << ";"

<< alpha \* 180 / M\_PI << ";" << betta \* 180 / M\_PI << ";"

<< Ksi \* 180 / M\_PI << ";" << hi \* 180 / M\_PI << ";" << rg << ";" << ENGINE\_WORKS << ";";

for (int i = 0; i < 8; i++) txtfile << engine\_done[i] << ";";

txtfile << endl;

cout << rg << endl;

cout << "END!" << endl;

system("pause");

return 0;

}

да

нет

Включение двигателя;  
Фиксация времени включения и номера двигателя;  
Расчет ky и kz

Рисунок В.1 – Блок-схема программы

да

да

да

да

да

да

нет

нет

нет

нет

нет

нет

нет

да

Данные для мины после сброса ГО

Интегрирование траектории

Par[i] = Par[i] + dPar[i]

Существует неиспользованный двигатель, для которого выполняется

Двигатель не работает  
и

Вне зоны ослепления

Двигатель работает

Выключение двигателя

Двигатель работает

и

Сброс ГО  
Изменение шага интегрирования

t > tГО

Начальные данные

Конец

Расчет промежуточных параметров на шаге

Пересчет ky и kz

y < 0

Интегрирование траектории

Par[i] = Par[i] + dPar[i]

На рисунке В.1 представлена блок-схема программы. Интегрирование траектории выполнялось методом Эйлера. Шаг интегрирования изначально принимается за 0.001 с. В блок-схеме опущена реализация дробления шага.

Массив Par[i] представляет собой массив интегрируемых параметров – время, координаты, скорости, угловые скорости, параметры Родрига-Гамильтона (всего 13 параметров). Массив dPar[i] представляет собой приращения соответствующего параметра на текущем шаге.

Время отделения головного обтекателя . Номинальное время полета .

После сброса ГО изменяется масса, характерная длина и аэродинамические характеристики мины. Шаг интегрирования принимается за 0.0001 с. Процесс реализации дробления шага также опущен.

За операции включения и выключения двигателя отвечает переменная ENGINE\_WORKS, она принимает только значения 0 (в данный момент ни один из двигателей не работает) или 1 (в данный момент включен один из двигателей). Массив переменных engine\_done[i] определяет, отработал ли двигатель, также принимает значения 0 и 1.

В блок «Расчет промежуточных параметров на шаге» входит расчет следующих параметров: углы, матрицы перехода, аэродинамические силы и моменты, текущий угол Пеленга, радиус-вектор до цели, текущий угол χ и другие. При интегрировании полета мины до сброса ГО блок расчета промежуточных параметров входит в блок «Интегрирование траектории», при этом не рассчитываются переменные, связанные с процессом корректировки (например, угол Пеленга).

Зона ослепления принимается за 50 метров.