1 Szenario

Gegeben sind der Ortsvektor $\vec{r_0} = (r_x, r_y, r_z)$ und der Geschwindigkeitsvektor $\vec{v_0} = (v_x, v_y, v_z)$ eines Objektes im Weltraum in ECI Koordinaten. Gesucht sind seine Bahnparameter. Es wird nun ein Verfahren angegeben, mit dem es möglich ist diese zu Bestimmen:

1. Bestimme den sowohl den Drehimpuls $\vec{h}$ und das Hamiltonsche Integral $\vec{q}$, als auch deren Beträge
2. Bestimme den Halbparameter p und die numerische Exzentrizität e.
3. Bestimme die Einheitsvektoren $\vec{e_z}$,$\vec{e_x}$,$\vec{e_y}$,$\vec{n}$, und $\vec{e_r}$.
4. Bestimme die Inklination i
5. Bestimme die Rekaszension Ω
6. Bestimme das Argument des Perigäums ω
7. Bestimme die wahre Anomalie θ_0
8. Bestimme den Perigäumsdurchgang t_p

Seien nun Beispielsweise die Anfangsbedingungen gegeben als:

$$\vec{r_0} = \begin{pmatrix} -37 \cdot 10^6 \\ 45 \cdot 10^6 \\ 38.5 \cdot 10^6 \end{pmatrix} [m]$$

$$\vec{v_0} = \begin{pmatrix} 3150 \\ -4830 \\ -2860 \end{pmatrix} \left[\frac{m}{s}\right]$$

2 Die 8 Schritte

In Schritt 1 gilt es den Drehimpuls und das Hamiltonsche Integral zu bestimmen, dies geht wie folgt:

$$\vec{h} = \vec{r_0} \times \vec{v_0}$$
$$\vec{q} = \vec{v_0} \times \vec{h} - \mu \frac{\vec{r_0}}{r_0}$$
$$\mu = 3,986 \cdot 10^{14} \left[\frac{m^3}{s^2}\right]$$

Dannach wird der Halbparameter und die numerische Exzentrizität bestimmt:

$$p = \frac{h^2}{\mu}$$
$$e = \frac{|\vec{q}|}{\mu}$$

Im 3 Schritt werden die Einheitsvektoren bestimmt :

$$\vec{e_z} = \frac{\vec{h}}{h}$$
$$\vec{e_x} = \frac{\vec{q}}{q}$$
$$\vec{e_y} = \vec{e_z} \times \vec{e_x}$$
$$\vec{n} = \frac{\vec{e_z^E} \times \vec{e_z}}{|\vec{e_z^E} \times \vec{e_z}|}$$
$$\vec{e_r} = \frac{\vec{r_0}}{r_0}$$
$$\vec{e_z^E} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \vec{e_y^E} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \vec{e_x^E} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

Es folgt im 4,5 und 6 Schritt die Bestimmung von i, Ω und ω:

$$i = \arccos(\vec{e_z} \cdot \vec{e_z^E})$$
$$\Omega = \left|\arctan 2\left(\vec{n} \cdot \vec{e_y^E}, \vec{n} \cdot \vec{e_x^E}\right)\right|$$
$$\omega = |\arctan 2\left(\vec{n} \cdot \vec{e_y}, \vec{n} \cdot \vec{e_x}\right)|$$

Im 7 Schritt wird die wahre Anomalie θ_0 bestimmt:

$$\theta_0 = \arctan 2\left(\vec{e_r} \cdot \vec{e_y}, \vec{e_r} \cdot \vec{e_x}\right)$$

Der 8 Schritt kann nun nicht mehr allgemein ausgeführt werden, denn je nach Wert der Exzentrizität e erhält man eine andere Bahnkurve:

1. $e > 1$ ergibt eine Hyperbel
2. $e = 1$ liefert eine Parabel
3. $e < 1$ resultiert in eine Ellipse
4. $e = 0$ ergibt einen Kreis als Spezialfall der Ellipse

Zu erreichen gilt es, über die Keplergleichung , die vergangene Zeit seit des Perigäumsdurchgangs t_p zu errechnen, dabei wird der Perigäumsdurchgang als $t_p = 0$ gesetzt zur einfacheren Rechnung der folgende Algorithmus deckt dabei alle Fälle der Bahnkurve ab: Eine 3-dimensionale Darstellung des

Algorithmus 1: Ein Algorithmus zum Errechnen der vergangenen Zeit seit des Perigäumsdurchgangs t_p für alle Bahnkurven

if $(e < 1)$ **then**
- $E = 2 \cdot \arctan\left(\sqrt{\frac{1-e}{1+e}} \cdot \tan\left(\frac{\theta_0}{2}\right)\right);$
- $\tau = E - e \sin E;$
- $a = \frac{p}{1-e^2};$
- $T = 2\pi\sqrt{\frac{a^3}{}\mu};$
- $t = -\frac{\tau \cdot T}{2\pi};$

if $(e = 1)$ **then**
- $u = 0.5 \cdot \tan\left(\frac{\theta_0}{2}\right);$
- $t = -\frac{\frac{u^3}{3}+u}{2\sqrt{\frac{\mu}{p^3}}};$

if $(e > 1)$ **then**
- $F = 2arc\tanh\left(\sqrt{\frac{e-1}{e+1}}\tan(0.5 \cdot \theta_0)\right);$
- $\tau = e \cdot \sinh(F) - F;$
- $a = \frac{p}{e^2-1};$
- $t = -\frac{\tau}{\sqrt{\frac{\mu}{a^3}}}$

Ortsvektors $\vec{r}$ erhält man anschließend auf folgende Weise:

$$\vec{r} = p \cdot \frac{\cos\theta \cdot e_x + \sin\theta \cdot e_y}{1 + e\cos\theta}$$

3 Umrechnung in Geodätische Koordinaten

Um Koordinaten von interialen ECI Koordinaten in sich mitrotierende ECEF Koordinaten zu kommen muss eine Rotationsmatrix (oder Quaternion, ...) um die z-Achse gebildet werden, die beide Koordinatensysteme ineinander transferiert. Da diese Transformation Zeitabhängig ist, ist eine Angabe eines Julianisches Datums(seit Jahr 2000) zur Umrechnung notwendig, um vom aktuellen Datum in das Julianische Datum zu kommen genügt folgendes:

$$d_{2000} = 367y - \left\lfloor \frac{7y + \lfloor \frac{m+9}{12} \rfloor}{4} \right\rfloor + \lfloor \frac{275m}{9} \rfloor + \frac{h + \frac{min}{60} + \frac{sek}{3600}}{24} + d - 730531.5$$

Hier setzt man wie folgt ein:

- Jahr y, zB 2012 für das Jahr 2012
- Monat m , zB 12 für den Dezember
- Tag d, zB 13 für den 13 des Monats
- Stunde h, zB 12 für 12 Uhr
- Minute min , zB 45 für 12:45
- Sekunde sek , zB 30,5 für 12:45:30 und 500ms , sek ist der einzige double Wert die anderen Parameter sind Integer

Die Rotationsmatrix um einen Vektor von ECI -Koordinaten nach ECF Koordinaten ist dann wie folgt bestimmt:

$$R_z(\phi) = \begin{pmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Den Winkel ϕ in Grad dafür erhält man wie folgt:

$$\phi = 280.46061837 + 360.98564736628 \cdot d_{2000}$$

Für die Anwendung in der Bibliothek ist darauf zu achten den Winkel vorher noch in Bogenmaß umzurechnen. Ist jetzt noch eine Umwandlung in geodätische Koordinaten (ϕ, λ, h) erwünscht, so müssen folgende Gleichungen nach ihnen aufgelöst werden:

$$
\begin{aligned}
x &= (N_\phi - h) \cdot \cos\phi \cdot \cos\lambda \\
y &= (N_\phi - h) \cdot \cos\phi \cdot \sin\lambda \\
z &= (N_\phi \cdot (1 - e^2) - h) \cdot \sin\phi \\
e &= 0.08181919 \quad , a = 6378137[m] \quad b = 6356752.3142[m] \\
N_\phi &= \frac{a}{\sqrt{1 - e^2 \cdot \sin^2\phi}}
\end{aligned}
$$

Dazu existiert keine geschlossene Lösung, eine Näherungslösung kann entweder iterativ oder mit folgendem Verfahren ermittelt werden:

Algorithmus 2: Ein Algorithmus zum Umrechnen von ECI in geodätische Koordinaten

---

Eingabe : Koordinaten im ECF (x, y, z)
Ausgabe : Koordinaten im geodätischen WGS84 Frame in (ϕ, λ, h)

$p = \sqrt{x^2 + y^2}$;
$E^2 = a^2 - b^2$;
$F = 54b^2z^2$;
$G = p^2 + (1 - e^2)z^2 - e^2E^2$;
$c = \frac{e^4Fp^2}{G^3}$;
$s = (1 + c + \sqrt{c^2 + 2c})^{\frac{1}{3}}$;
$P = \frac{F}{3(s + \frac{1}{s} + 1)^2G^2}$;
$Q = \sqrt{1 + 2e^4P}$;
$r_0 = -\frac{Pe^2p}{1+Q} + \sqrt{0.5a^2(1 + \frac{1}{Q}) - \frac{P(1-e^2)z^2}{Q(1+Q)} - 0.5Pp^2}$;
$U = \sqrt{(p - e^2r_0)^2 + z^2}$;
$V = \sqrt{(p - e^2r_0)^2 + (1 - e^2)z^2}$;
$z_0 = \frac{b^2z}{aV}$;
$e' = e\frac{a}{b}$;
$\phi = \arctan\left(\frac{z + (e')^2z_0}{p}\right) \cdot \frac{180}{\pi}$;
$D = 42.111 - 25.4119\phi + 8.716449\phi^2 - 0.0646199\phi^3$;
$\lambda = \arctan 2(y, x) \cdot \frac{180}{\pi}$;
$h = U\left(\frac{b^2}{aV} - 1\right) - D$

Dieser Algorithmus macht je nach Breitengrad ϕ einen Fehler bis maximal 50 Meter in der Höhe h.

4 Implementierung

Zu obiger Angabe von $\vec{r_0}$ und $\vec{v_0}$ wird nun hier der Code zum berechnen der 8 Schritte angegeben:

```
double rx = -37e6;
double ry = 45e6;
double rz = 38.5e6;
double mu = 3.986e14;
double Re = 6378137;

double vx = 3150;
double vy = -4830;
double vz = -2860;

Vector3D r0(rx,ry,rz);
Vector3D v0(vx,vy,vz);

Vector3D exE(1,0,0);
Vector3D eyE(0,1,0);
Vector3D ezE(0,0,1);

Vector3D h_vec = r0.cross(v0);
double h = h_vec.getLength();

Vector3D q_vec;
q_vec = v0.cross(h_vec).vecSub(r0.normalize().scale(mu));
double q = q_vec.getLength();

double p = h*h/mu;
double e = q/mu;

Vector3D ez =h_vec.normalize();
Vector3D ex = q_vec.normalize();
Vector3D ey =ez.cross(ex);
Vector3D n;
n = ezE.cross(ez).scale(1/(ezE.cross(ez).getLength()));
Vector3D er = r0.normalize();
```

```
double i = acos(ez.dot(ezE));
double Omega = atan2(n.dot(eyE),(n.dot(exE)));
double omega = acos(n.dot(ex));
double theta0 = atan2(er.dot(ey),er.dot(ex));

// e>1 Fall

double F = 2*atanh(sqrt((e-1)/(e+1))*tan(0.5*theta0));
double tau = e*sinh(F)-F;
double a = p/(e*e-1);
double t = -tau/(sqrt(mu/(a*a*a)));

// Berechne Perigaeumsabstand

double rp =a*(e-1);

// Berechne wahre Anomalie des Aufpralls

double theta_knall = -acos(1/e*(p/Re -1));

//Berechne verbleibende Zeit

double F_knall;
F_knall = 2*atanh(sqrt((e-1)/(e+1))*tan(0.5*theta_knall));
double tau_knall = e*sinh(F_knall)-F_knall;
double t_knall = -tau_knall/(sqrt(mu/(a*a*a)));

//Berechne Aufprallort in ECI-Koordinaten

double ot = omega + theta_knall;
double rx_knall;
double ry_knall;
double rz_knall;
rx_knall = cos(ot)*cos(Omega)-sin(ot)*sin(Omega)*cos(i);
ry_knall = cos(ot)*sin(Omega)+sin(ot)*cos(Omega)*cos(i);
rz_knall = sin(ot)*sin(i);


Vector3D ECI_knall(rx_knall,ry_knall,rz_knall);
ECI_knall = ECI_knall.scale(p/(1+e*cos(theta_knall)));
```

```
// ECI nach ECF nach geodetic am 13.3.7 um 0:00:00

double y2k = daysSinceY2k(2007,3,13,0,0,0);
Matrix3D toECF = eciToECF(y2k);
Vector3D ecf_knall = ECI_knall.matVecMult(toECF);
Vector3D geodetic = ecfToGeodetic(ecf_knall);
```

In diesem zugegeben etwas länglichen Code wird folgendes gemacht:

1. die Bahnparameter berechnet.
2. geprüft, ob ein Einschlag auf der Erde stattfindet
3. den Einschlagsort in ECI und die Einschlagszeit berechnen.
4. für den fiktiven Einschlagszeitpunkt 13.3.7 um 0:00:00 (es kann jeder Zeitpunkt gewählt werden) Die Koordinaten in (ϕ, λ, h) berechnen.

Die Ausgabe dieses kleinen Programms liefert:
Exzentrizität e = 1.386530 => Hyperbel;
Inklination i = 58.06 Grad
Ω = 105.105 Grad
ω = 167.006 Grad
θ_0= -126.49 Grad
t = 9489.95 Sekunden
Perigäumsabstand = 5133169 < R_E => Aufprall unvermeidlich
Remaining Time bis zum Aufprall Δt = 9119.74 Sekunden
Aufprall bei $(\phi, \lambda) = (48.35, 51.24)$ Grad
Der Einschlagsort befindet sich an der Grenze zwischen Russland und Kasachstan