The Stark problem

The Stark problem is a gravitational one-body problem with an external force field constant both in magnitude and direction. It is also known as the *low-thrust* and *accelerated Kepler problem*. The Hamiltonian in cartesian coordinates can be formulated as:

$$\mathcal{H}(\boldsymbol{v};\boldsymbol{r}) = \frac{1}{2}v^2 - \frac{\mu}{r} - \varepsilon z.$$

The constant force field is usually oriented towards the positive z axis to exploit the cylindrical symmetry of the problem.

The Stark problem is Liouville integrable, but the exact solution involves parabolic coordinates, a Sundman transformation and the use of elliptic functions. For small values of the constant force field ε, a perturbative treatment yields a solution which provides a better dynamical and physical insight.

Action angle variables

We are going to employ an Hamiltonian perturbation scheme based on Lie series, which requires the Hamiltonian to be expressed in a set of action-angle variables for the unperturbed problem. In our case, the Hamiltonian is split as

$$\mathcal{H} = \mathcal{H}_0 + \varepsilon\mathcal{H}_1,$$

with

$$\mathcal{H}_0 = \frac{1}{2}v^2 - \frac{\mu}{r}$$

as the unperturbed problem and

$$\mathcal{H}_1 = -z$$

as the perturbation. A commonly used set of action-angle variables for the unperturbed problem is the set of *Delaunay* orbital elements, which are related to the classical Keplerian elements by the following relations:

$$L = \sqrt{\mu a} \qquad G = L\sqrt{1-e^2} \qquad H = G\cos i$$

for the momenta, and

$$l = M \qquad g = \omega \qquad h = \Omega$$

for the coordinates. In this new coordinate system, $\mathcal{H}_0$ assumes a simple form:

$$\mathcal{H}_0 = -\frac{\mu^2}{2L^2}.$$

In order to express $\mathcal{H}_1$ (i.e., z) in this new coordinate system, we first start from the expression of the position vector in the orbital plane:

$$\left(r\cos f, r\sin f, 0\right),$$

and then we apply a composite rotation matrix and extract the z component from the position vector, yielding

$$z = r\cos f\sin i\sin g + r\sin f\sin i\cos g.$$

We proceed then to express

$$\cos i = \frac{H}{G}, \qquad \sin i = \sqrt{1-\cos^2 i} = \sqrt{1-\frac{H^2}{G^2}}.$$

We have seen in the previous section how to develop r/a, $\cos f$ and $\sin f$ in terms of e and M (where $M \equiv l$). We thus need only to express e and a in terms of Delaunay elements. Straightforwardly:

$$e = \sqrt{1-\frac{G^2}{L^2}}$$

and

$$a = \frac{L^2}{\mu}.$$

We are now ready to proceed with the implementation. Before starting, we copy the implementations of the required elliptic expansions from the previous lecture:

In [1]:
```
from pyranhapp0x import *
from fractions import Fraction as Frac

settings.max_term_output = 5

pst = poisson_series.get_type('polynomial_rational')

def besselJ(n,x,order):
    retval = 0
    for m in range(0,int((order-n)/2) + 1):
```

```
        retval = retval + Frac((-1)**m,math.factorial(m)*math.factorial(m+n)) * (x/2)**(2*m+n)
    return retval

def r_a(e,M,order):
    retval = 1 + Frac(1,2) * e**2
    for s in range(1,order + 1):
        retval = retval - e * Frac(1,s) * besselJ(s-1,s*e,order - 1) * math.cos(s*M)
    for s in range(1,order - 1):
        retval = retval + e * Frac(1,s) * besselJ(s+1,s*e,order - 1) * math.cos(s*M)
    return retval

def sin_f(e,M,order):
    retval = 0
    for s in range(1,order + 2):
        retval = retval + besselJ(s-1,s*e,order) * math.sin(s*M)
    for s in range(1,order):
        retval = retval - besselJ(s+1,s*e,order) * math.sin(s*M)
    retval = retval * binomial_exp(1,-e**2,Frac(1,2),int(order/2))
    return retval.transform(lambda t: (t[0].filter(lambda u: u[1].degree(['e']) <= order),t[1]))

def cos_f(e,M,order):
    retval = 0
    for s in range(1,order + 2):
        retval = retval + besselJ(s-1,s*e,order) * math.cos(s*M)
    for s in range(1,order):
        retval = retval + besselJ(s+1,s*e,order) * math.cos(s*M)
    retval = retval * (1 - e**2) - e
    return retval.transform(lambda t: (t[0].filter(lambda u: u[1].degree(['e']) <= order),t[1]))

def binomial_exp(x,y,r,order):
    retval = 0
    for k in range(0,order + 1):
        retval = retval + math.binomial(r,k) * x**(r-k) * y**k
    return retval
```

Implementation

As a first step, we define the truncation order of our perturbation theory, i.e., the maximum degree of the variables in terms of which the elliptic expansions are performed:

In [2]:
```
order = 4
```

Then we start defining some useful variables:

In [3]:
```
e,l,L,mu,g,si,G,H,eps = pst('e'),pst('l'),pst('L'),pst(r'\mu'),pst('g'),pst('s_i'),pst('G'),pst('H'),pst(r'\varepsilon')
```

Recalling now that

$\mathcal{H}_1 = -z = -r\cos f \sin i \sin g - r \sin f \sin i \cos g,$

we can write for $\mathcal{H}_1$:

In [4]:
```
HH1 = -r_a(e,l,order) * L**2 * mu**-1 * cos_f(e,l,order) * si * math.sin(g) -\
    r_a(e,l,order) * L**2 * mu**-1 * sin_f(e,l,order) * si * math.cos(g)
HH1
```

Out[4]:

$$\left(-\frac{65}{384}\frac{L^2e^6s_i}{\mu} + \frac{619}{1152}\frac{L^2e^8s_i}{\mu} - \frac{L^2s_i}{\mu} + \frac{1}{2}\frac{L^2e^2s_i}{\mu} + \frac{1}{64}\frac{L^2e^4s_i}{\mu}\right)\sin\left(g+l\right) + \left(\frac{3}{2}\frac{L^2es_i}{\mu} - \frac{5}{384}\frac{L^2e^5s_i}{\mu} + \frac{1667}{18432}\frac{L^2e^7s_i}{\mu}\right)\sin\left(g\right) -$$

Here we have to remember that e is to be regarded as a function of G and L via the relation:

$e = \sqrt{1 - \frac{G^2}{L^2}},$

and s_I as a function of H and G via the relation

$s_i = \sin i = \sqrt{1 - \frac{H^2}{G^2}}.$

We can now proceed to compute the normal form. At the first order, the normal form corresponds to an averaging procedure, which will remove from the perturbation all those terms depending harmonically on l. In the general case we can compute the averaging by an explicit integration, but in this specific case it will be enough to filter out all terms not depending on the fast variable l from the perturbation.

In [5]:
```
HH1_av = HH1.filter(lambda t: t[1].t_order(['l']) == 0)
HH1_av
```

Out[5]: $$\left(\frac{3}{2}\frac{L^2 e s_i}{\mu} - \frac{5}{384}\frac{L^2 e^5 s_i}{\mu} + \frac{1667}{18432}\frac{L^2 e^7 s_i}{\mu}\right)\sin\left(g\right)$$

The normal form is thus of the type:

$$\mathcal{H} = \mathcal{H}_0 + \varepsilon\mathcal{F}_0 \sin g.$$

If we now want to derive the equations of motion, we need to keep in mind the fact that e and s_i are implicit functions of the Delaunay variables. Piranha can be instructed to keep in mind these explicit dependencies when computing partial derivatives. For instance, for the partial derivative with respect to G we will have:

$$\frac{\partial}{\partial G} = \frac{\partial}{\partial e}\frac{\partial e}{\partial G} + \frac{\partial}{\partial s_i}\frac{\partial s_i}{\partial G}.$$

Given any piranha series type, we can register *custom derivatives* as follows:

In [6]:
```
pst.register_custom_derivative('G',lambda s: s.partial('e') * (-G * L**-2 * e**-1) + s.partial('s_i') * (si**-1 * H**2 * G*
```

In [7]:
```
pst.register_custom_derivative('H',lambda s: s.partial('s_i') * (-H * G**-2 * si**-1))
```

In [8]:
```
pst.register_custom_derivative('L',lambda s: s.partial('L') + s.partial('e') * (G**2 * L**-3 * e**-1))
```

We can now generate the equations of motion:

In [9]:
```
HH = -mu**2 * L**-2 / 2 + eps * HH1_av
```

$\frac{dL}{dt}$:

In [10]:
```
-math.partial(HH,'l')
```

Out[10]: 0

$\frac{dG}{dt}$:

In [11]:
```
-math.partial(HH,'g')
```

Out[11]: $$\left(\frac{5}{384}\frac{L^2 \varepsilon e^5 s_i}{\mu} - \frac{1667}{18432}\frac{L^2 \varepsilon e^7 s_i}{\mu} - \frac{3}{2}\frac{L^2 \varepsilon e s_i}{\mu}\right)\cos\left(g\right)$$

$\frac{dH}{dt}$:

In [12]:
```
-math.partial(HH,'h')
```

Out[12]: 0

$\frac{dl}{dt}$:

In [13]:
```
math.partial(HH,'L')
```

Out[13]: $$\frac{\mu^2}{L^3} + \left(-\frac{5}{192}\frac{L\varepsilon e^5 s_i}{\mu} - \frac{25}{384}\frac{G^2 \varepsilon e^3 s_i}{L\mu} + \frac{1667}{9216}\frac{L\varepsilon e^7 s_i}{\mu} + \frac{11669}{18432}\frac{G^2 \varepsilon e^5 s_i}{L\mu} + 3\frac{L\varepsilon e s_i}{\mu} + \ldots\right)\sin\left(g\right)$$

$\frac{dg}{dt}$:

In [14]:
```
math.partial(HH,'G')
```

Out[14]: $$\left(-\frac{3}{2}\frac{G\varepsilon s_i}{\mu e} + \frac{3}{2}\frac{H^2 L^2 \varepsilon e}{G^3 \mu s_i} + \frac{1667}{18432}\frac{H^2 L^2 \varepsilon e^7}{G^3 \mu s_i} - \frac{11669}{18432}\frac{G\varepsilon e^5 s_i}{\mu} - \frac{5}{384}\frac{H^2 L^2 \varepsilon e^5}{G^3 \mu s_i} + \ldots\right)\sin\left(g\right)$$

$\frac{dh}{dt}$:

In [15]:
```
math.partial(HH,'H')
```

Out[15]:

$$\left(\frac{5}{384}\frac{HL^{2}\varepsilon e^{5}}{G^{2}\mu s_{i}}-\frac{3}{2}\frac{HL^{2}\varepsilon e}{G^{2}\mu s_{i}}-\frac{1667}{18432}\frac{HL^{2}\varepsilon e^{7}}{G^{2}\mu s_{i}}\right)\sin\left(g\right)$$