New Orbits for the Equimass n-Body Problem

Robert J. Vanderbei—The LOQO Guy

May 14, 2004
Workshop on
Large Scale Nonlinear and Semidefinite Programming
Waterloo, Ontario

Operations Research and Financial Engineering, Princeton University
http://www.princeton.edu/∼rvdb

Optimization

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & b \leq h(x) \leq b + r, \\ & l \leq x \leq u \end{array}$$

- *Linear Programming (LP):* f and h are linear.
- *Convex Optimization:* f is convex, each h_i is concave, and $r = \infty$.
- *Nonlinear Optimization:* f and each h_i is assumed to be twice differentiable

- Generally, we seek a *local solution* in the vicinity of a given starting point.
- If problem is convex (which includes LP), any local solution is automatically a *global solution*.

Least Action Principle

Given: n bodies.

Let:
m_j denote the mass and
$z_j(t)$ denote the position in $\mathbb{R}^2 = \mathbb{C}$ of body j at time t.

Action Functional:

$$A = \int_0^{2\pi} \left(\sum_j \frac{m_j}{2} \|\dot{z}_j\|^2 + \sum_{j,k:k<j} \frac{m_j m_k}{\|z_j - z_k\|} \right) dt.$$

Equation of Motion

First Variation:

$$\begin{aligned}\delta A &= \int_0^{2\pi} \sum_\alpha \left(\sum_j m_j \dot{z}_j^\alpha \delta\dot{z}_j^\alpha - \sum_{j,k:k<j} m_j m_k \frac{(z_j^\alpha - z_k^\alpha)(\delta z_j^\alpha - \delta z_k^\alpha)}{\|z_j - z_k\|^3} \right) dt \\ &= -\int_0^{2\pi} \sum_j \sum_\alpha \left(m_j \ddot{z}_j^\alpha + \sum_{k:k\neq j} m_j m_k \frac{z_j^\alpha - z_k^\alpha}{\|z_j - z_k\|^3} \right) \delta z_j^\alpha dt\end{aligned}$$

Setting first variation to zero, we get:

$$m_j \ddot{z}_j^\alpha = -\sum_{k:k\neq j} m_j m_k \frac{z_j^\alpha - z_k^\alpha}{\|z_j - z_k\|^3}, \qquad j = 1, 2, \ldots, n, \quad \alpha = 1, 2$$

Note: If $m_j = 0$ for some j, then the first order optimality condition reduces to $0 = 0$, which is *not* the equation of motion for a massless body.

Periodic Solutions

We assume solutions can be expressed in the form

$$z_j(t) = \sum_{k=-\infty}^{\infty} \gamma_k e^{ikt}, \qquad \gamma_k \in \mathbb{C}.$$

Writing with components $z_j(t) = (x_j(t), y_j(t))$ and $\gamma_k = (\alpha_k, \beta_k)$, we get

$$\begin{aligned} x(t) &= a_0 + \sum_{k=1}^{\infty} \left(a_k^c \cos(kt) + a_k^s \sin(kt) \right) \\ y(t) &= b_0 + \sum_{k=1}^{\infty} \left(b_k^c \cos(kt) + b_k^s \sin(kt) \right) \end{aligned}$$

where

$$\begin{array}{lll} a_0 = \alpha_0, & a_k^c = \alpha_k + \alpha_{-k}, & a_k^s = \beta_{-k} - \beta_k, \\ b_0 = \beta_0, & b_k^c = \beta_k + \beta_{-k}, & b_k^s = \alpha_k - \alpha_{-k}. \end{array}$$

The variables a_0, a_k^c, a_k^s, b_0, b_k^c, and b_k^s are the decision variables in the optimization model.

The AMPL Model

```
param N := 3;  # number of masses
param n := 15;  # number of terms in Fourier series representation
param m := 100;  # number of terms in numerical approx to integral

param theta {j in 0..m-1} := j*2*pi/m;

param a0 {i in 0..N-1} default 0;          param b0 {i in 0..N-1} default 0;
var as {i in 0..N-1, k in 1..n} := 0;      var bs {i in 0..N-1, k in 1..n} := 0;
var ac {i in 0..N-1, k in 1..n} := 0;      var bc {i in 0..N-1, k in 1..n} := 0;

var x {i in 0..N-1, j in 0..m-1}
  = a0[i]+sum {k in 1..n} ( as[i,k]*sin(k*theta[j]) + ac[i,k]*cos(k*theta[j]) );
var y {i in 0..N-1, j in 0..m-1}
  = b0[i]+sum {k in 1..n} ( bs[i,k]*sin(k*theta[j]) + bc[i,k]*cos(k*theta[j]) );

var xdot {i in 0..N-1, j in 0..m-1}
  = if (j<m-1) then (x[i,j+1]-x[i,j])*m/(2*pi) else (x[i,0]-x[i,m-1])*m/(2*pi);
var ydot {i in 0..N-1, j in 0..m-1}
  = if (j<m-1) then (y[i,j+1]-y[i,j])*m/(2*pi) else (y[i,0]-y[i,m-1])*m/(2*pi);

var K {j in 0..m-1} = 0.5*sum {i in 0..N-1} (xdot[i,j]^2 + ydot[i,j]^2);

var P {j in 0..m-1}
  = - sum {i in 0..N-1, ii in 0..N-1: ii>i}
          1/sqrt((x[i,j]-x[ii,j])^2 + (y[i,j]-y[ii,j])^2);

minimize A: (2*pi/m)*sum {j in 0..m-1} (K[j] - P[j]);
```

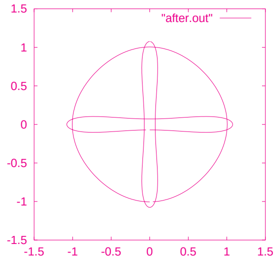
Continued...

```
let {i in 0..N-1, k in 1..n} as[i,k] := 1*(Uniform01()-0.5);
let {i in 0..N-1, k in 1..n} ac[i,k] := 1*(Uniform01()-0.5);
let {i in 0..N-1, k in n..n} bs[i,k] := 0.01*(Uniform01()-0.5);
let {i in 0..N-1, k in n..n} bc[i,k] := 0.01*(Uniform01()-0.5);

solve;
```

Choreographies and the Ducati

The previous AMPL model was used to find many *choreographies* (a la Moore and Montgomery/Chencinier) in the equimass n-body problem and the stable *Ducati* solution to the 3-body problem.



Contents