ORF 522: Lecture 18

Nonconvex Optimization: Models

Robert J. Vanderbei

November 21, 2013

Slides last edited at 11:34am on Thursday 5th December, 2013

Operations Research and Financial Engineering, Princeton University

http://www.princeton.edu/∼rvdb

Examples: Nonconvex Optimization Models

Celestial Mechanics—Periodic Orbits

- Find periodic orbits for the planar gravitational n-body problem.
- Minimize action:

$$\int_0^{2\pi} (K(t) - P(t))dt,$$

- where $K(t)$ is kinetic energy,

$$K(t) = \frac{1}{2}\sum_i \left(\dot{x}_i^2(t) + \dot{y}_i^2(t)\right),$$

- and $P(t)$ is potential energy,

$$P(t) = -\sum_{i<j} \frac{1}{\sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2}}.$$

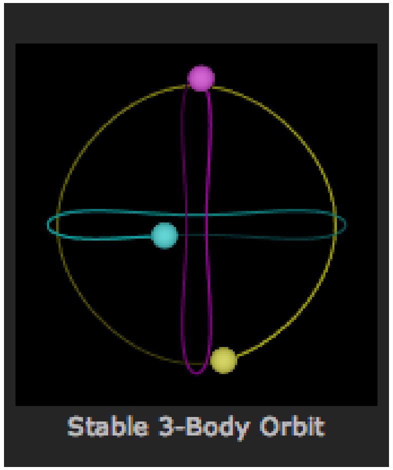
- Subject to periodicity constraints:

$$x_i(2\pi) = x_i(0), \qquad y_i(2\pi) = y_i(0).$$

Specific Example

Orbits.mod with $n = 3$ and $(0, 2\pi)$ discretized into a 160 pieces gives the following results:

constraints	0	
variables	960	
time (secs)		
LOQO	1.1	
LANCELOT	8.7	
SNOPT	287	(no change for last 80% of iterations)



AMPL Model

```
param N := 3;  # num. of masses
param n := 30;  # num. of terms in Fourier series
param m := 300;  # num. of terms in num. approx to integral

param pi := 4*atan(1);
param t {j in 0..m-1} := j*2*pi/m;

var as {i in 0..N-1, k in 1..n};
var ac {i in 0..N-1, k in 1..n};

var bs {i in 0..N-1, k in 1..n};
var bc {i in 0..N-1, k in 1..n};

var cs {i in 0..N-1, k in 1..n};
var cc {i in 0..N-1, k in 1..n};

# fixing these to zero helps loqo converge
var a0 {i in 0..N-1} default 0;
var b0 {i in 0..N-1} default 0;
var c0 {i in 0..N-1} default 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*t[j]) +
                ac[i,k]*cos(k*t[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*t[j]) +
                bc[i,k]*cos(k*t[j]) );

var z {i in 0..N-1, j in 0..m-1}
        = c0[i]+sum {k in 1..n} (
                cs[i,k]*sin(k*t[j]) +
                cc[i,k]*cos(k*t[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 zdot {i in 0..N-1, j in 0..m-1}
        = if (j<m-1) then (z[i,j+1]-z[i,j])*m/(2*pi)
                     else (z[i,0]-z[i,m-1])*m/(2*pi);

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

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

let {i in 0..N-1} a0[i] := 0;

# This makes the planar Ducati
let ac[0,1] := 1;
let bs[1,1] := 1;
let ac[2,1] := -1;
let bs[2,1] := -1;

solve;
```

Putting on an Uneven Green

Given:

- $z(x, y)$ elevation of the green.
- Starting position of the ball (x_0, y_0).
- Position of hole (x_f, y_f).
- Coefficient of friction μ.

Find: initial velocity vector so that ball will roll to the hole and arrive with minimal speed.

Variables:

- $u(t) = (x(t), y(t), z(t))$—position as a function of time t.
- $v(t) = (v_x(t), v_y(t), v_z(t))$—velocity.
- $a(t) = (a_x(t), a_y(t), a_z(t))$—acceleration.
- T—time at which ball arrives at hole.

See http://www.boeing.com/boeing/phantom/socs/putting.page
and http://orfe.princeton.edu/ rvdb/ampl/nlmodels/puttputt/golfvis.pdf

Putting—Two Approaches

1. Problem can be formulated with two decision variables:

   $$v_x(0) \quad \text{and} \quad v_y(0)$$

   and two constraints:

   $$x(T) = x_f \quad \text{and} \quad y(T) = y_f.$$

   In this case, $x(T)$, $y(T)$, and the objective function T are complicated functions of the two variables that can only be computed by integrating the appropriate differential equation.

2. A discretization of the complete trajectory (including position, velocity, and acceleration) can be taken as optimization variables and the physical laws encoded in the differential equation can be written as constraints.

The first approach requires an optimization algorithm that does not use derivatives.

Interior-point methods require first and second derivatives.

Optimization methods not requiring derivatives are called *derivative-free* methods.

Such methods are *slow* and *imprecise* but have become *extremely popular*, because many people are simply too lazy to follow the second approach.

Putting—Continued

Objective:

$$\text{minimize } v_x(T)^2 + v_y(T)^2.$$

Constraints:

$$\begin{aligned} v &= \dot{u} \\ a &= \dot{v} \\ ma &= N + F - mge_z \\ u(0) &= u_0 \qquad u(T) = u_f, \end{aligned}$$

where

- m is the mass of the golf ball.
- g is the acceleration due to gravity.
- e_z is a unit vector in the positive z direction.

and ...

Putting—Continued

- $N = (N_x, N_y, N_z)$ is the normal force:

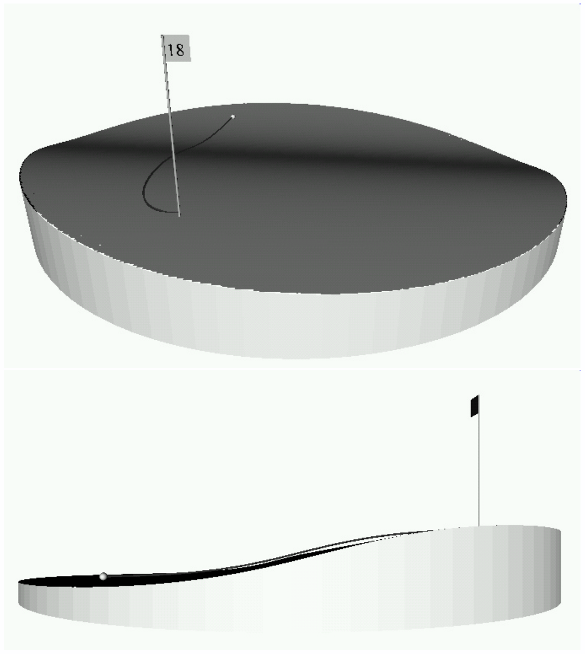
$$\begin{aligned} N_z &= m\frac{g - a_x(t)\frac{\partial z}{\partial x} - a_y(t)\frac{\partial z}{\partial y} + a_z(t)}{(\frac{\partial z}{\partial x})^2 + (\frac{\partial z}{\partial y})^2 + 1} \\ N_x &= -\frac{\partial z}{\partial x} N_z \\ N_y &= -\frac{\partial z}{\partial y} N_z. \end{aligned}$$

- F is the force due to friction:

$$F = -\mu \|N\| \frac{v}{\|v\|}.$$

Putting—Specific Example

- Discretize continuous time into $n = 200$ discrete time points.
- Use finite differences to approximate the derivatives.



constraints	597
variables	399
time (secs)	
LOQO	14.1
LANCELOT	> 600.0
SNOPT	4.1

AMPL Model

```
param g := 9.8;  # acc due to gravity
param m := 0.0459; # mass of a golf ball (in kilograms)

param x0 :=  1; # coords of starting point
param y0 :=  2;

param xn :=  1; # coords of ending point
param yn := -2;

param n := 500; # number of discete time steps
param mu;       # coefficient of friction

var x{0..n};  # coordinates of the trajectory
var y{0..n};
var T >= 0;   # total time for the putt

# Here we define the elevation of the green
var z   {i in 0..n} = -0.3*atan(x[i]+y[i]);
var dzdx{i in 0..n} = -0.3/(1+(x[i]+y[i])^2);
var dzdy{i in 0..n} = -0.3/(1+(x[i]+y[i])^2);

# The velocity vector.  v[i] denotes the derivative at the
# midpoint of the interval i(T/n) to (i+1)(T/n).
var vx{i in 0..n-1} = (x[i+1]-x[i])*n/T;
var vy{i in 0..n-1} = (y[i+1]-y[i])*n/T;
var vz{i in 0..n-1} = (z[i+1]-z[i])*n/T;

# The acceleration vector.  a[i] denotes the accel at the midpt
# of the interval (i-0.5)(T/n) to (i+0.5)(T/n), i.e. at i(T/n).
var ax{i in 1..n-1} = (vx[i]-vx[i-1])*n/T;
var ay{i in 1..n-1} = (vy[i]-vy[i-1])*n/T;
var az{i in 1..n-1} = (vz[i]-vz[i-1])*n/T;

var Nz{i in 1..n-1};
var Nx{i in 1..n-1} = -dzdx[i]*Nz[i];
var Ny{i in 1..n-1} = -dzdy[i]*Nz[i];
var Nmag{i in 1..n-1} = sqrt(Nx[i]^2 + Ny[i]^2 + Nz[i]^2);
```

```
var vx_avg{i in 1..n-1} = (vx[i]+vx[i-1])/2;
var vy_avg{i in 1..n-1} = (vy[i]+vy[i-1])/2;
var vz_avg{i in 1..n-1} = (vz[i]+vz[i-1])/2;
var speed{i in 1..n-1}
      = sqrt(vx_avg[i]^2 + vy_avg[i]^2 + vz_avg[i]^2);

var Frx{i in 1..n-1} = -mu*Nmag[i]*vx_avg[i]/speed[i];
var Fry{i in 1..n-1} = -mu*Nmag[i]*vy_avg[i]/speed[i];
var Frz{i in 1..n-1} = -mu*Nmag[i]*vz_avg[i]/speed[i];

minimize finalspeed: vx[n-1]^2 + vy[n-1]^2;

s.t. newt_x {i in 1..n-1}: ax[i]=(Nx[i]+Frx[i])/m;
s.t. newt_y {i in 1..n-1}: ay[i]=(Ny[i]+Fry[i])/m;
s.t. newt_z {i in 1..n-1}: az[i]=(Nz[i]+Frz[i]-m*g)/m;

s.t. xinit: x[0] = x0;
s.t. yinit: y[0] = y0;

s.t. xfinal: x[n] = xn;
s.t. yfinal: y[n] = yn;

s.t. onthegreen {i in 0..n}: x[i]^2+y[i]^2 <= 16;

let T := 5;
let mu := 0.0;

let {i in 0..n} x[i]  := (i/n)*xn + (1-i/n)*x0;
let {i in 0..n} y[i]  := (i/n)*yn + (1-i/n)*y0;

option loqo_options "verbose=2 inftol=5e-5";
solve;

let mu := 0.05;
solve;
```

Goddard Rocket Problem

Objective:

maximize $h(T)$;

Constraints:

$$
\begin{aligned}
v &= \dot{h} \\
a &= \dot{v} \\
\theta &= -c\dot{m} \\
ma &= (\theta - \sigma v^2 e^{-h/h_0}) - gm \\
0 &\leq \theta \leq \theta_{\max} \\
m &\geq m_{\min} \\
h(0) &= 0 \qquad v(0) = 0 \qquad m(0) = 3
\end{aligned}
$$

where

- $\theta =$ Thrust , $m =$ mass
- $\theta_{\max}$, g, σ, c, and h_0 are given constants
- h, v, a, T_h, and m are functions of time $0 \leq t \leq T$.

Goddard Rocket Problem—Solution

constraints	399
variables	599
time (secs)	
LOQO	5.2
LANCELOT	(IL)
SNOPT	(IL)