Optimization with ODE'S constraints and mesh independence

Vitor Leite Nunes

Department of Mathematics
Interdisciplinary Center for Applied Mathematics
Virginia Tech

vitor@vt.edu

Problem Formulation

- Minimize the following functional:

$$\min_{(x,a)} f(x,a)$$

$$C(x^n, x^{n-1}, \ldots, x, t, a) = 0$$

$$t \in \left[0, T\right]$$

Constructing the Lagrangian

- Since the constraints are true for every $t \in [0, T]$
- It holds

$$L(x, a, \lambda) = f(x, a) - \sum_{i \in I} \int_0^T \lambda_i(t) * C_i(x, a) dt$$

Numerical Approximation

- After setting the Lagrangian we apply the KKT conditions and introduce an numerical algorithm. This algorithm varies from problem to problem.
- Basically solving $\nabla L = 0$ trough a numerical algorithm.

Mesh Independence Principle for a Newton Iteration (MIP)

- Given a fixed staring point the number of iterations to reach a fixed tolerance is eventually independent of the size of the mesh, h.

Sufficient Conditions for MIP (E.L. Allgower and K. Bohmer):

- If:

1) Let F be the infinite dimensional problem to solve F(x)=0

$$F'(z_n)c_n = -F(z_n)$$

$$z_{n+1} = c_n + z_n$$

$$n = 0,1,...$$

$$\| z_{n+1} - z_n \| \leq k \| z_n - z_{n-1} \|^2$$

Sufficient Conditions for MIP (E.L. Allgower and K. Bohmer):

- If :

2) The Lipchitz-continuity is verified for ϕ^h (discrete form of the problem)

3) Δ^h is equibounded (the mesh)

4) The scheme $\phi^h{}'(\Delta^h z)$ is stable.

5) Consistency for F(z)

6) And F'(v)w smooth enough

We have mesh independence

Case Study

$$\min_{(a,b)} \int_0^T g(\dot{x})dt + \varepsilon_1 * a^2 + \varepsilon_2 * b^2$$

(1)

$$m\ddot{x} = f(x)$$

$$x(0) = a;$$

$$x(T) = b;$$

$$\varepsilon_1 > 0, \varepsilon_2 > 0$$

Lagrangian Associated to the Problem

- The correspondent Lagrangian is :

$$L(x,y,a,b,\lambda_1,\lambda_2,\mu_1,\mu_2) = \int_0^T g(y)dt + \varepsilon_1 * a^2 + \varepsilon_2 * b^2 +$$

$$+\int_0^T \lambda_1(t) * (m\dot{y} - f(x))\,dt + \int_0^T \lambda_2(t) * (y - \dot{x})\,dt + \mu_1(x(0) - a) + \mu_2 * (x(T) - b)$$

KKT Conditions

- It is obtained the next 12 equations:

$$x(0) = a$$
$$x(T) = b$$
$$m\dot{y} = f(x)$$
$$y = \dot{x}$$
$$\lambda_1(0) = 0$$
$$\lambda_1(T) = 0$$
$$\dot{\lambda}_2(t) + \lambda_1(t) * f'(x) = 0$$
$$g'(y) + \lambda_2(t) - m\dot{\lambda}_1(t) = 0$$
$$\mu_1 = -\lambda_2(0)$$
$$\mu_2 = \lambda_2(T)$$
$$a = \mu_1 / (2\varepsilon_1)$$
$$b = \mu_2 / (2\varepsilon_2)$$

An Algorithm is Born

- Looking carefully, a natural fix point algorithm is born:
- Given an (a,b) it is possible to solve :

$$x(0) = a$$
$$x(T) = b$$
$$m\dot{y} = f(x)$$
$$y = \dot{x}$$

- which makes possible the resolution of:

$$\lambda_1(0) = 0$$
$$\lambda_1(T) = 0$$
$$\dot{\lambda}_2(t) + \lambda_1(t) * f'(x) = 0$$
$$g'(y) + \lambda_2(t) - m\dot{\lambda}_1(t) = 0$$

An Algorithm is Born

- Then solving :

$$\mu_1 = -\lambda_2(0)$$
$$\mu_2 = \lambda_2(T)$$
$$a = \mu_1 /(2\varepsilon_1)$$
$$b = \mu_2 /(2\varepsilon_2)$$

- We obtain the new pair (a,b), which it can be plugged again in the first system making an iterative procedure.

Convergence

- *The pair obtained is a descent direction*
- Proof:
- The gradient of L is

$$\nabla_{(a,b)} L = (2a\varepsilon_1 - \mu_1, 2b\varepsilon_2 - \mu_2)$$

- and the Hessian is given by :

$$H_{(a,b)} = \begin{pmatrix} 2\varepsilon_1 & 0 \\ 0 & 2\varepsilon_2 \end{pmatrix}$$

Convergence

- So we have :

$$H_{(a,b)}^{-1} = \begin{pmatrix} 1/2\varepsilon_1 & 0 \\ 0 & 1/2b\varepsilon_2 \end{pmatrix}$$

- Doing :

$$-H_{(a,b)}^{-1} * \nabla_{(a,b)} L = \begin{pmatrix} \mu_1/(2\varepsilon_1) - a \\ \mu_2/(2\varepsilon_2) - b \end{pmatrix} = (a_{k+1}, b_{k+1}) - (a_k, b_k)$$

- So it is a descendent direction. Another way to see it, is that H is define positive so will always be a descendent direction.

ODE’S Numerical Approximation

- To solve the ODE’S systems was use the finite element method.
- Which approximates the solution by piecewise polynomials.

Numerical Results for the Example:

$$g(\dot{x}) = \sqrt{1+\dot{x}^2}$$

$$f(x) = x+1$$

$$\min_{(a,b)} \int_0^T \sqrt{1+\dot{x}^2}\,dt + \varepsilon_1 * a^2 + \varepsilon_2 * b^2$$

$$m\ddot{x} = x+1$$

$$x(0) = a$$

$$x(T) = b$$

Numerical Results

$a_0 = -1$

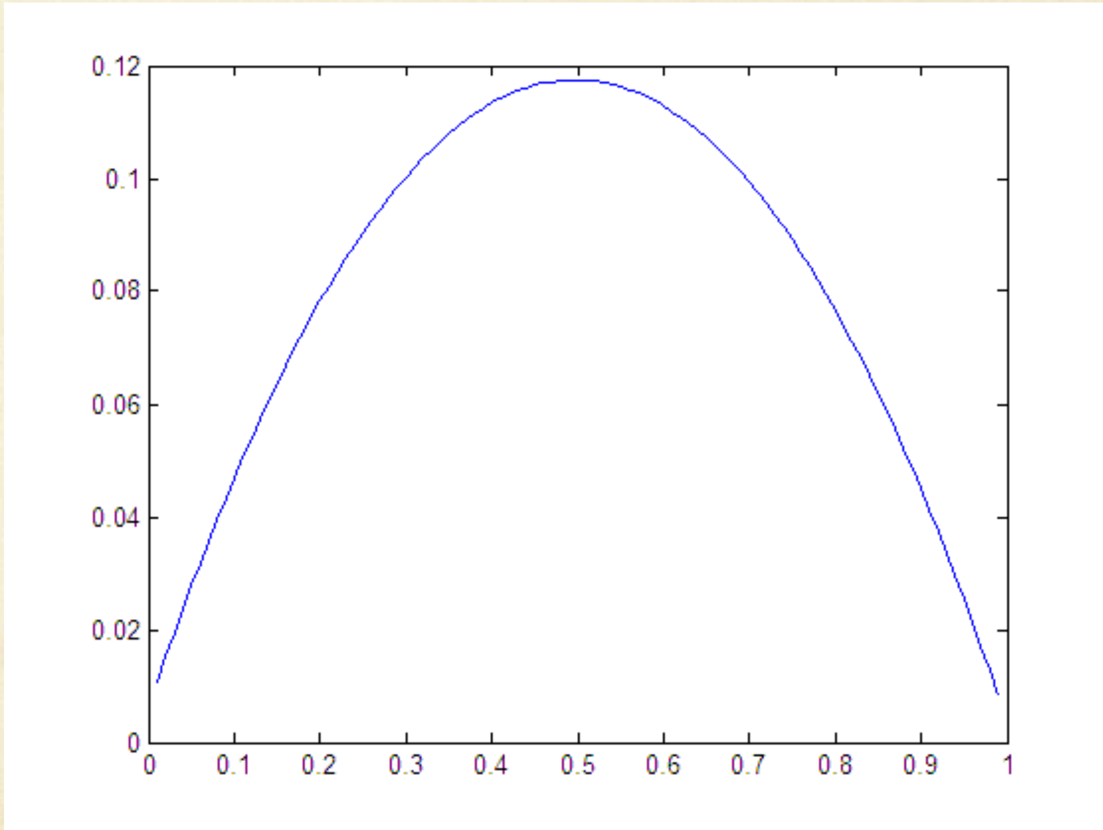
$b_0 = 1$

$\varepsilon_1 = 3$

$\varepsilon_2 = 3$

$control = 10^{-4}$

$control = \| (a_{k+1}, b_{k+1}) - (a_{k+1}, b_{k+1}) \|_2$



Numerical Results

h (mesh size)	0.1	0.02	0.01	0.001
Iterations	9	9	9	9
a	0.0146	0.0070	0.0062	0.0054
b	-0.0123	0.0018	0.0036	0.0052

Conclusions

- The problem is very sensitive for the changing of the $\varepsilon's$.
- The convergence, and its rate depends on the $\varepsilon's$.
- When the $\varepsilon's$ are very small the algorithm has difficulties to converge.
- There is mesh independence.

OBRIGADO

Thanks