CasADi tutorials Lund, December 2011

Exercise 3: Nonlinear programming using IPOPT

Joel Andersson

Optimization in Engineering Center (OPTEC) and ESAT-SCD, K.U. Leuven,
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium

December 5, 2011

1 NLP Solvers in CasADi

The NLP solvers interfaced with CasADi solves NLP:s of the following form:

$$\begin{array}{cc} \underset{x \in \mathbb{R}^n}{\text{minimize:}} & f(x) \\ \text{subject to:} & \begin{array}{c} x_{\min} \le x \le x_{\max} \\ g_{\min} \le g(x) \le g_{\max} \end{array} \end{array} \tag{1}$$

With the functions f and g formulated as the CasADi functions `f` and `g`, an NLP solver instance can be allocated by:

```
nlp_solver = IpoptSolver(f,g)
```

The interface will then automatically generate the information that it might need to solve the NLP, which may be solver and option dependent. Typically an NLP solver will need a function that gives the Jacobian of the constraint function and a Hessian of the Lagrangian function ($L(x, \lambda) = f(x) + \lambda^{\mathrm{T}} g(x)$) with respect to x. The interface will generate this kind of information automatically and provide to the solver. For the IPOPT interface, if the user wants to use a particular Hessian approximation (for example a Gauss-Newton Hessian), this is possble by simply prividing a third argument to the constructor:

```
nlp_solver = IpoptSolver(f,g,h)
```

NLP solvers, like ODE/DAE integrators are functions in CasADi (though taking derivatives of them is currently not supported). You will find the input and output schemes in the CasADi API documentation on the website or by using the question mark in Python.

2 Exercises

3.1 Formulate and solve the Rosenbrock problem:

$$\begin{array}{cc} \underset{x,y,z \in \mathbb{R}}{\text{minimize:}} & x^2 + 100\, z^2 \\ \text{subject to:} & z + (1-x)^2 = y \end{array} \tag{2}$$

Use $x = 2.5$, $y = 3.0$, $z = 0.75$, as a starting point. How many iterations do you need to to converge using default options?

3.2 By default, the IPOPT interface will use a BFGS approximation. You can have CasADi generating an exact Hessian instead using the command:

```
solver.setOption("generate_hessian",True)
```

How does this influence the number of iterations?

3.3 A function might have multiple local minima. Consider the function:

$$f(x, y) = \exp(-x^2 - y^2)\ \sin(4\,(x + y + x * y^2)) \tag{3}$$

in the domain $[0, 1] \times [0, 1]$. You can visualize the function it using the following lines in Python:

```
import numpy as NP
from matplotlib import pylab as plt

# Domain
x = NP.linspace(-1,1,100)
y = NP.linspace(-1,1,100)
[X,Y] = plt.meshgrid(x,y)

# Function
F = NP.exp(-X**2-Y**2)*NP.sin(4*(X+Y+X*Y**2))

# Plot the function
plt.clf()
plt.contour(X,Y,F)
plt.colorbar()
plt.jet()
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

Find the unconstrained minimizer of the function starting at different starting points, e.g. $[0, 0]$, $[0.9, 0, 9]$, $[-0.9, -0, 9]$. What do you see? To solve unconstrained problems with CasADi, simply leave out the second argument to `IpoptSolver`.

3.4 **Extra:** Consider a nonlinear pendulum given by the dynamics:

$$\dot{p} = v \tag{4}$$

$$\dot{v} = u - C\ \sin(\frac{p}{C}) \tag{5}$$

with $C = 18/\pi$. Define the state vector $x = [p, v]^{\mathrm{T}}$.

Use the RK4 integrator from exercise 2,

$$
\begin{aligned}
k_1 &= f(x_k, u_k) && (6)\\
k_2 &= f(x_k + \frac{1}{2}\,\Delta t\, k_1, u_k) && (7)\\
k_3 &= f(x_k + \frac{1}{2}\,\Delta t\, k_2, u_k) && (8)\\
k_4 &= f(x_k + \Delta t\, k_3, u_k) && (9)\\
x_{k+1} &= x_k + \frac{1}{6}\,\Delta t\,(k_1 + 2\,k_2 + 2\,k_3 + k_4) && (10)
\end{aligned}
$$

to define a discrete time system of the form $x_{k+1} = \Phi(x_k, u_k)$ (only one RK4 step per control interval).

Choose the initial values $p_0 = 10$, $v_0 = 0$ and take $N = 50$ time steps, each of size $\Delta t = 0.2$. Constrain p and v to the interval $[-10, 10]$ and u to the interval $[-3, 3]$. This allows us to formulate the following discrete time optimal control problem:

$$
\begin{aligned}
\underset{x_0, u_0, x_1, \ldots, u_{N-1}, x_N}{\text{minimize:}} \quad & \sum_{k=0}^{N-1} \|u_k\|_2^2 \\
\text{subject to:} \quad & [p_0, v_0]^{\mathrm{T}} - x_0 = 0 \\
& \Phi(x_k, u_k) - x_{k+1} = 0 \qquad \text{for } k = 0, \ldots, N-1 \\
& x_N = 0 \\
& -10 \le x_k \le 10 \qquad \text{for } k = 0, \ldots, N-1 \\
& -10 \le u_k \le 10 \qquad \text{for } k = 0, \ldots, N-1
\end{aligned} \qquad (11)
$$

Solve the problem using IPOPT and plot the optimal solution.