5. opti

Hands-on CasADi course on optimal control

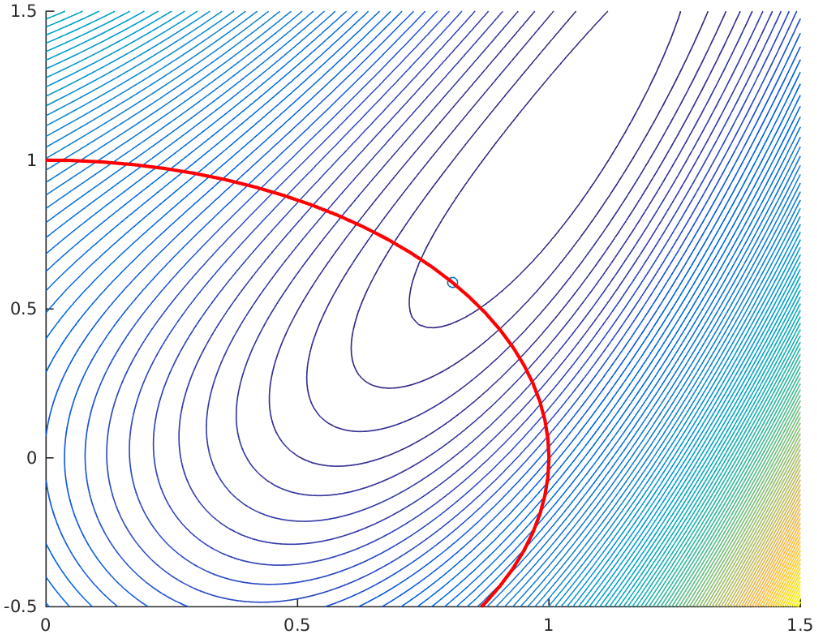
Joris Gillis

Goal: from paper to computer

$$\underset{x,y}{minimize}\quad (1-x)^2+(y-x^2)^2$$
$$s.t.\quad x^2+y^2=1$$

```
opti = casadi.Opti();

x = opti.variable();
y = opti.variable();

opti.minimize((1-x)^2+(y-x^2)^2);
opti.subject_to(x^2+y^2==1);

opti.solver('ipopt');
sol = opti.solve();

sol.value([x y])
```

```
ans =
    0.8082    0.5889
```



Contents

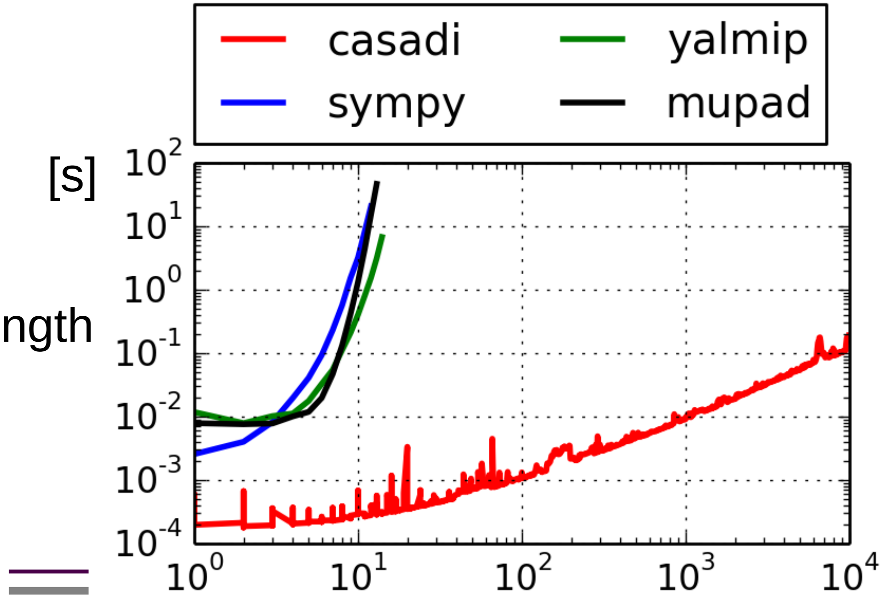
- Goal
- **Why?**
- How?
- Help? (What if my NLP does not solve?)

Why CasADi?

- C++11 core; Python, Matlab/Octave interfaces
- C code generation
- Comes with solver interfaces
- Scales well (matrix-valued expression graphs)
- Algorithmic differentiation (AD) performs better than symbolic differentiation

Illustration:

Gradient construction time versus horizon length (single-shooting optimal control problem)



Contents

- Goal
- Why?
- **How?**
- Help? (What if my NLP does not solve?)

How?

- Creating symbols for decision variables

```
opti = casadi.Opti();

x_mat  = opti.variable(4,5);
x_vec  = opti.variable(6);
x_scal = opti.variable();
```

Under the hood: MX.sym

How?

- Composing expressions from symbols

```
expr = cos(0.1*x_scal)
```

```
expr =

cos((0.1*opti1_x_3))
```

```
hessian(expr,x_scal)
```

```
ans =

@1=0.1, (-(@1*(0.1*cos((@1*opti1_x_3)))))
```

How?

- Specify an expression to minimize

```
opti = casadi.Opti();

x = opti.variable(10);
y = opti.variable();

opti.minimize(cos(sum(10*x)));

opti
```

```
opti =

Opti {
  instance #0
  #variables: 1 (nx = 10)
  #parameters: 0 (np = 0)
  #constraints: 0 (ng = 0)
  CasADi solver needs updating.
}
```

How?

- Add constraints

```
% scalar
opti.subject_to(y==3);
% vector
opti.subject_to([x(end);y]>=[0;2]);
```

```
opti =

Opti {
  instance #1
  #variables: 2 (nx = 11)
  #parameters: 0 (np = 0)
  #constraints: 2 (ng = 3)
  CasADi solver needs updating.
}
```

How?

- Clear constraints

```
opti.subject_to();
```

How?

- Solve

```
opti.solver('ipopt',options);
sol = opti.solve();
```

```
Number of nonzeros in equality constraint Jacobian...:        1
Number of nonzeros in inequality constraint Jacobian.:        2
Number of nonzeros in Lagrangian Hessian.............:       55

iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
   0  1.0000000e+00 3.00e+00 5.00e-01  -1.0 0.00e+00    -  0.00e+00 0.00e+00   0
   1  9.9813891e-01 0.00e+00 3.00e+04  -1.0 3.00e+00   4.0 2.20e-02 1.00e+00f  1
...
  13 -1.0000000e+00 0.00e+00 1.65e-08  -8.6 1.10e-07  -0.8 1.00e+00 1.00e+00f  1
  14 -1.0000000e+00 0.00e+00 6.04e-09  -9.0 1.20e-07  -1.3 1.00e+00 1.00e+00f  1

Number of Iterations....: 14

                                   (scaled)                 (unscaled)
Objective...............:  -1.0000000000000000e+00   -1.0000000000000000e+00
Dual infeasibility......:   6.0368021307602012e-09    6.0368021307602012e-09
Constraint violation....:   0.0000000000000000e+00    0.0000000000000000e+00
Complementarity.........:   9.0909090909090889e-10    9.0909090909090889e-10
Overall NLP error.......:   6.0368021307602012e-09    6.0368021307602012e-09

EXIT: Optimal Solution Found.
```

How?

- Retrieve solution

```
ysol = sol.value(y)
sol.value([sum(x) sqrt(y)])
```

```
ysol =

     3


ans =

    0.3142    1.7321
```

How?

- Supply initial guess

```
opti = casadi.Opti();
x = opti.variable();

opti.minimize(sin(x)^2);
opti.solver('ipopt');

sol = opti.solve();
sol.value(x)
```

```
ans =

     0
```

```
opti.set_initial(x, 6);
sol = opti.solve();
sol.value(x)
```

```
ans =

     6.2832
```

How?

- Parametric NLP

```
opti = casadi.Opti();
x = opti.variable();
p = opti.parameter();

opti.subject_to(sin(x)==p);
opti.solver('ipopt');

opti.set_value(p, 0.1);
sol = opti.solve();
sol.value(x)
```

```
ans =

    0.1002
```

```
opti.set_value(p, 0.2);
sol = opti.solve();
sol.value(x)
```

```
ans =

    0.2014
```

How?

- Inspect standard form

```
sol.value(jacobian(opti.g,opti.x))
```

```
ans =


   (2,10)        1
   (1,11)        1
   (3,11)        1
```

$$\underset{x}{\text{minimize}} \quad f(x,p)$$

$$s.t. \quad g_{LB} \leq g(x,p) \leq g_{UB}$$

Contents

- Goal
- Why?
- How?
- **Help?** (What if my NLP does not solve?)

Infeasible problem

```
opti = casadi.Opti();

x = opti.variable();
y = opti.variable();

opti.subject_to(-3<=x<= 3);
opti.subject_to(y>=0);
opti.subject_to(sin(x*y)>=2);

opti.solver('ipopt');
sol = opti.solve();
```

```
Error using casadi.Opti/solve (line 239)
Error in Opti::solve [OptiNode] at .../casadi/core/optistack.cpp:159:
.../casadi/core/optistack_internal.cpp:943: Assertion "return_success()" failed:
Solver failed. You may use opti.debug.value to investigate the latest values of
variables. return_status is 'Infeasible_Problem_Detected'

Error in demo (line 34)
sol = opti.solve();
```

Infeasible problem

```
opti.debug.show_infeasibilities
```

```
Violated constraints (tol 0), in order of declaration:
------- i = 3/3 ------
2 <= 1 <= inf (viol 1)
Opti constraint of shape 1x1, defined at
/home/yacoda/leuven2018/5.opti/demo.m:31 in demo
  opti.subject_to(sin(x*y)>=2);
```

Invalid number

```
opti = casadi.Opti();

x = opti.variable();
y = opti.variable();

opti.minimize((x-2)^2);
opti.subject_to(y>=sqrt(x));

opti.solver('ipopt');
sol = opti.solve();
```

```
CasADi - 2018-05-11 12:04:14 WARNING("solver:nlp_jac_g failed: Inf detected for
output jac_g_x, at (row 1, col 1).") [.../casadi/core/oracle_function.cpp:249]
Error evaluating Jacobian of inequality constraints at user provided starting
point.

Number of Iterations....: 0

EXIT: Invalid number in NLP function or derivative detected.

Error using casadi.Opti/solve (line 239)
Error in Opti::solve [OptiNode] at .../casadi/core/optistack.cpp:159:
.../casadi/core/optistack_internal.cpp:943: Assertion "return_success()" failed:
Solver failed. You may use opti.debug.value to investigate the latest values of
variables. return_status is 'Invalid_Number_Detected'
```

Invalid number

```
CasADi - 2018-05-11 12:04:14 WARNING("solver:nlp_jac_g failed: Inf detected for
output jac_g_x, at (row 1, col 1).") [.../casadi/core/oracle_function.cpp:249]
```

```
opti.debug.g_describe(1)
opti.debug.x_describe(1)
```

```
ans =

Opti constraint of shape 1x1, defined at
/home/jgillis/conferences/linkoping/5.opti/demo.m:44 in demo
  opti.subject_to(y>=sqrt(x));

ans =

Opti decision variable 'opti19_x_1' of shape 1x1, defined at
/home/jgillis/conferences/linkoping/5.opti/demo.m:40 in demo
  x = opti.variable();
```

Invalid number

$$\frac{d\sqrt{x}}{dx} = \frac{1}{2\sqrt{x}}$$

Invalid number

```
opti = casadi.Opti();

x = opti.variable();
y = opti.variable();

opti.minimize((x-2)^2);
opti.subject_to(y>=sqrt(x));

opti.set_initial(x,1);

opti.solver('ipopt');
sol = opti.solve();
```

5. opti