Prof. Dr. Kathrin Flaßkamp
M.Sc. Matthias Hoffmann
Systems Modeling and Simulation
Saarland University

Optimization – CasADi Cheat Sheet – WS 21/22

`import casadi.*`	-	Import CasADi into your project, can be done first, if you do not want to type `casadi.` in front of CasADi functions.
`opt = casadi.Opti()`	-	Initialize an opti object that collects your decision variables, parameters, constraints, cost functions, settings, etc.
`opt.solver('ipopt')`	-	Set the solver for your problem in `opt` to IPOPT, other solvers like WORHP are also possible.
`dec = opt.variable(n,m)`	-	Generate a symbolic decision variable matrix with size $n \times m$ for your problem, e.g. the inputs u in an optimal control problem.
`par = opt.parameter(n, m)`	-	Generate a symbolic parameter matrix with size $n \times m$ for your problem, e.g. the Pareto parameters of a MOO problem. Before solving of a problem, these parameters have to be replaced with numerical values with the `set_value` method.
`opt.set_value(par, num)`	-	Set the value of parameter `par` to num.
`opt.minimize(cost)`	-	Sets the cost function to be `cost`. `cost` has to depend on your decision variables.
`opt.subject_to(input)`	-	Define a constraint for the optimization problem. `input` is typically a logical expression, e.g. $0 \leq$ `dec` ≤ 1. If no input is given, all constraints are deleted.
`opt.set_initial(dec, num)`	-	Set `num` as the initial guess for the decision variable `dec`.
`sol = opt.solve()`	-	Solve the optimization problem in `opt` and generate a solution object `sol` that stores all the information of the corresponding optimization process. The sol objects can also be stored in a solution matrix `sol(i)= opt.solve()`.
`x_val = sol.value(x)`	-	Store the computed value of `x` in `x_val`.
`var = casadi.MX(n,m)`	-	Generate an empty $n \times m$ matrix that you want to fill with values, for example the k-matrix in Runge-Kutta integrators. Do **not** use opti-variables for this as this can cause problems in the optimization.