# Tutorial 1: Introduction to Optimization

## GIAN Short Course on Optimization: Applications, Algorithms, and Computation

Sven Leyffer

Argonne National Laboratory

September 12-24, 2016

# Create Your Software Environment

1. Download the AMPL trial version for your machine, from `wiki.mcs.anl.gov/leyffer/` under Courses & Lectures
2. Dowbload and install the AMPL mode for `emacs` or `vim`, see `github.com/dpo/ampl-mode/blob/master/emacs/ampl-mode.el`
3. Download the COIN-OR solvers, and install them, see `projects.coin-or.org/CoinBinary`
   - Requires `make`, `c++`, `fortran`
   - Run script `get.AllThirdParty` to get missing files
   - Add binaries to your path ... e.g. `/bin`
4. Test it: Download the AMPL models from Lecture 1, and run.

# Theory Questions

1. Solve the problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \; c^T x \quad \text{subject to } l \leq x \leq u,$$

where $c, l, u \in \mathbb{R}^n$ and $-\infty < l \leq u < \infty$.
What happens if some $l_i$ or $u_i$ are not finite?

# AMPL Model for Optimal Reinforcement

1. Write an AMPL model of the reinforced concrete beam example:

$$
\begin{array}{lll}
\underset{x}{\text{minimize}} & f(x) = 29.4x_1 + 0.6x_2x_3 & \text{cost of beam} \\
\text{subject to} & c(x) = x_1x_2 - 7.735\frac{x_1^2}{x_2} \geq 180 & \text{load constraint} \\
& x_3 - 4x_2 \geq 0 & \text{width/depth ratio} \\
& 40 \leq x_1 \leq 77,\ x_2 \geq 0,\ x_3 \geq 0 & \text{simple bounds,}
\end{array}
$$

where

- $x_1$ = area of re-inforcement, e.g. $x_1 \in \{40, 45, \ldots, 75\}$
- $x_2$ = width of beam,
- $x_3$ = depth of beam.

# AMPL Model for Optimal Technology Penetration

Write an AMPL model of the basic technology transition problem

$$\begin{aligned}
\underset{\{q^o,q^n,x,z\}(t)}{\text{maximize}} \quad & \int_0^T e^{-rt}\left[\tilde{S}(q^o(t)+q^n(t),t) - c_o q^o(t) - c_n(x(t))q^n(t)\right] dt \\
\text{subject to} \quad & \dot{x}(t) = q^n(t), \qquad x(0) = x_0 = 0 \\
& \dot{z}(t) = e^{-at}\big(b_o q^o(t) + b_n q^n(t)\big), \qquad z(0) = z_0 = 0 \\
& z(T) \le z_T \\
& q^o(t) \ge 0, \qquad q^n(t) \ge 0.
\end{aligned}$$

where

$$\tilde{S}(Q,t) = e^{bt} S(Qe^{-bt}),$$

where $b > 0$ growth rate of demand, and

$$S(Q) = \begin{cases} S(Q) = S_0 \ln Q, & \text{if } \sigma = 1 \\ \frac{S_0}{1-\sigma} Q^{1-\sigma}, & \text{otherwise,} \end{cases}$$

# AMPL Model for Optimal Technology Penetration

| Parameter | Unit | Notation | Value |
|---|---|---|---|
| Discount rate | - | $r$ | 0.05 |
| Demand exponent | - | $\sigma$ | 2.0 |
| Demand scale | \$B | $S_0$ | 98,000 |
| Demand growth rate | - | $b$ | 0.015 |
| Environmental rate | - | $a$ | 0.02 |
| Emissions, old tech. | tC/mBTU | $b_o$ | 0.02 |
| Emissions, new tech. | tC/mBTU | $b_n$ | 0.001 |
| Unconstrained emissions | BtC | $Z_{\max}$ | 61.9358 |
| Emission reduction % | - | $\zeta$ | 0.5 |
| Emissions target | BtC | $z_T = \zeta Z_{\max}$ | |
| Production cost, old tech. | \$/mBTU | $c_o$ | 20 |
| Starting cost, new tech. | \$/mBTU | $c_n^0$ | 50 |
| Learning rate | - | $\gamma$ | 0.85 |
| Initial experience | quad | $x_0$ | 0 |
| Experience unit size | quad | $\bar{X}$ | 300 |

# AMPL Model for Optimal Technology Penetration

Write an AMPL model of the basic technology transition problem

$$\underset{\{q^o,q^n,x,z\}(t)}{\text{maximize}} \int_0^T e^{-rt}\left[\tilde{S}(q^o(t)+q^n(t),t)-c_o q^o(t)-c_n(x(t))q^n(t)\right]dt$$

$$\begin{aligned}
\text{subject to}\quad & \dot{x}(t)=q^n(t), \qquad x(0)=x_0=0\\
& \dot{z}(t)=e^{-at}\big(b_o q^o(t)+b_n q^n(t)\big), \qquad z(0)=z_0=0\\
& z(T)\le z_T\\
& q^o(t)\ge 0, \qquad q^n(t)\ge 0.
\end{aligned}$$

1. Solve the model for different time steps, $h$
2. Experiment with different discretization schemes, e.g. explicit or implicit Euler and Trapezoidal rule
3. Use AMPL's `fprintf` to output your results & plot the transition paths in Matlab.

# Optimal Technology Penetration: Hints & AMPL Tricks

## Attributes & Defined Variables

```
param Ntime > 0, integer, default 200;  # ... time-steps

set T := 0..Ntime;

var qo{T} >= 0, := 10;          # ... old technology
var qn{T} >= 0, := 1;           # ... new technology

var z{T};                       # ... emissions

var dzdt{t in T: t<Ntime} = (z[t+1] - z[t])/h;

subject to

 # ... ODE for z(t)
 InitZ: z[0] = z0;                  # ... initial cond.
 DiffEqnZ{t in T: t<Ntime}:         # ... disc. ODE
  dzdt[t] = exp(-a*t*Tend/Ntime) * (bo*qo[t] + bn*qn[t]);
```