Optimization and Control of Chemical Processes

Nuno M.C. Oliveira
<http://www.eq.uc.pt/~nuno/cim2003/>

GEPSI — PSE Group, Department of Chemical Engineering
University of Coimbra, Portugal

Workshop on Modeling and Simulation in Chemical Engineering
June 30 - July 4, 2003

In memoriam.
José Almiro A.M. Castro (1953–2002)

Outline

- An overview of **PSE**.
- **Numerical methods**:
  - **Nonlinear equations**.
  - **Continuous** optimization
  - Combining **discrete** decisions, **logical** and **heuristic** information.
- Major **application areas** of PSE:
  - **Design**.
  - **Operations**.
  - **Control**.
- **Software tools**.

1. An overview of Process Systems Engineering

The essence of Chemical Engineering has always been

> "the **synthesis**, **design**, **testing**, **scaleup**, **operation**, **control** and **optimization** of processes that change the physical state or the composition of materials" (Westerberg, 1998).

PSE has traditionally been concerned with

> "understanding and developing **systematic procedures** for **design**, **control** and **operation** of chemical processes" (Sargent, 1991).

PSE can be identified with a ***major paradigm*** in Chemical Engineering.

1.1. Major paradigms in Chemical Engineering:

① **unit operations** (A.D. Little, 1915).

② basis on **physical sciences** and **mathematical analysis**:

- Aris, Bird et al. (2002), $\sim$ 60s.

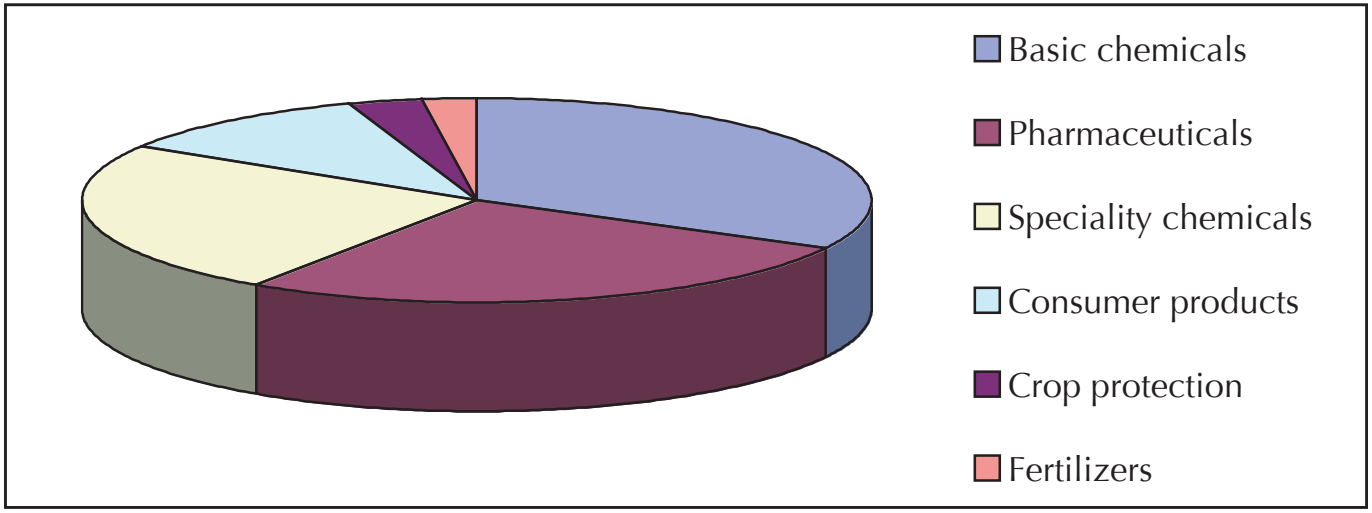
③ **systematic decision making** methods for design and operation:

- volume in *AIChE Symposium Series* (1961).
- Rudd and Watson (1968).
- journal *Computers and Chemical Engineering*, (1977–).
- Douglas (1988)
- Biegler et al. (1997)
- ESCAPE meetings (1992–)
  - ▷ ***ESCAPE-14, Lisbon, May 16–19, 2004***, see <http://www.escape14.online.pt>

⇨ **PSE area is 35–40 years old.**

1.2. Basic ChemE economic data (US, 2001)

Distribution of revenues

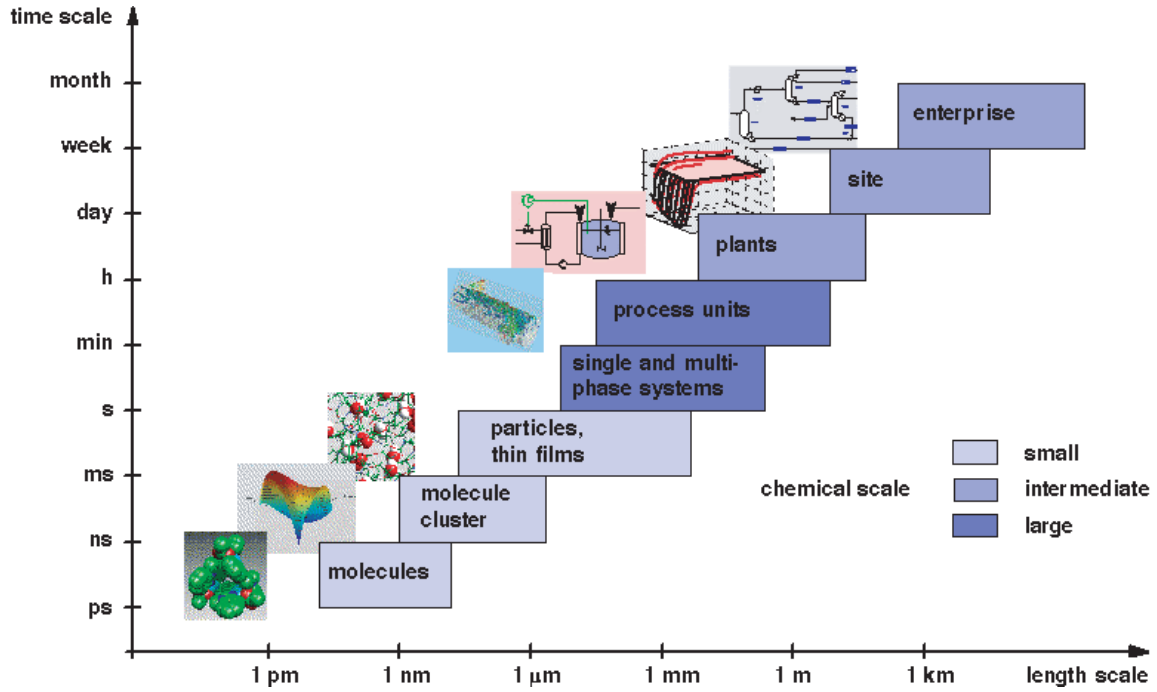


Total revenues: Chemicals (447 bil.) + Oil (595 bil.) $\simeq$ **1 000 billion USD**.

Typical profit margins (Grossmann, 2004):

Biotechnology:	20–30%
Pharmaceutical:	15–20%
Petroleum:	6 – 10%
Chemicals:	5 – 8%

The concept of PSE has recently evolved to include improvement of the decision making process of the entire **chemical supply chain** — **creation** and **operation**:



- **Micro-scale:**
  - ▷ **Product engineering**: ***synthesize*** new products (solvents, refrigerants, polymers), shorter development phases, agile manufacturing.
- **Macro-scale:**
  - ▷ **Enterprise level:** ***logistics*** for manufacturing, production planning and distribution.

➪ **Discover**, **design**, **manufacture** and **distribute** chemical products in the context of many **conflicting goals**.

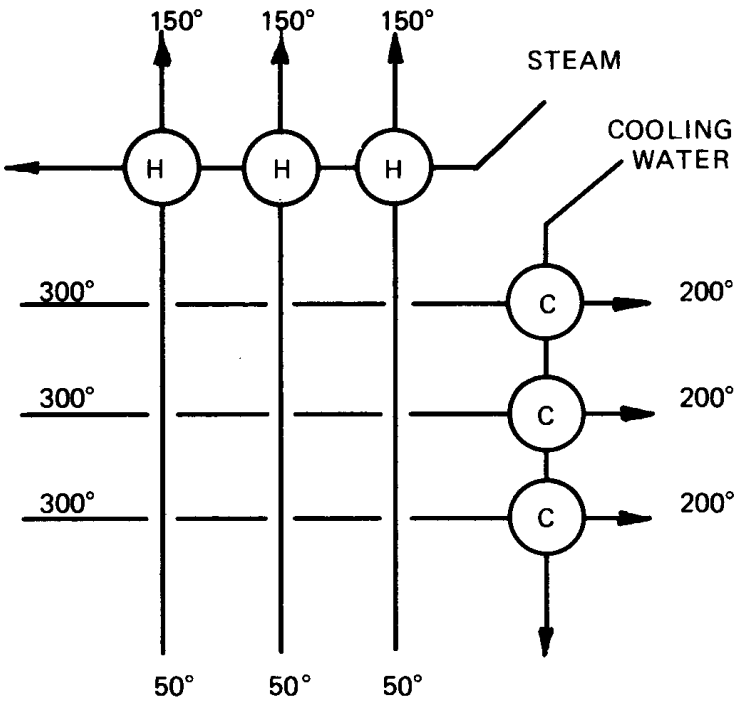
Important characteristics of decision making methodologies:

- essential to compete in **global markets**.
  - ▷ reduce costs, operate efficiently, environmental safe and sustainable, innovative products, improved quality.
- with both **practical** and **theoretical** implications.
  - ▷ while largely driven by industrial needs, address fundamental theoretical aspects.
- integrate **physical sciences**, **mathematics**, **operations research** (& A.I.), and **computer science**.
  - ▷ models often without closed form solutions.
  - ▷ efficient problem representations for alternatives.
  - ▷ product is often a (complex) computer code.

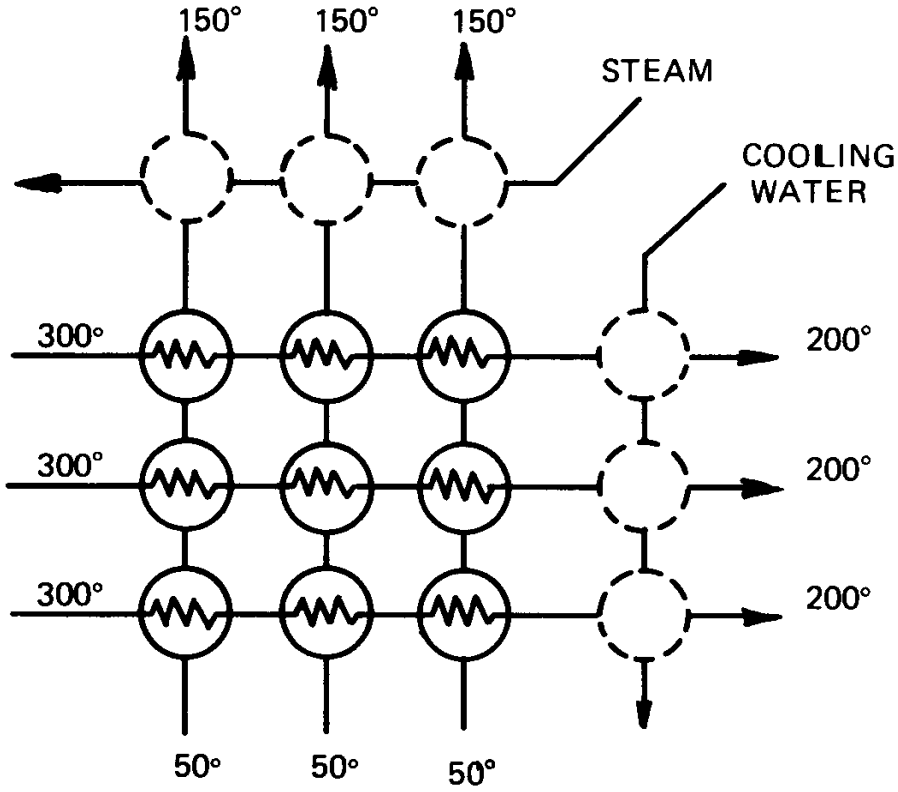
Representations and **models** are the natural tools to generate reasonable alternatives. Only systematic methods for exploring alternatives guarantee solutions that **meet constraints** and **optimize** an objective.

1.3. A simple example: Alternative heat transfer networks

Extreme configurations:



Non-integrated

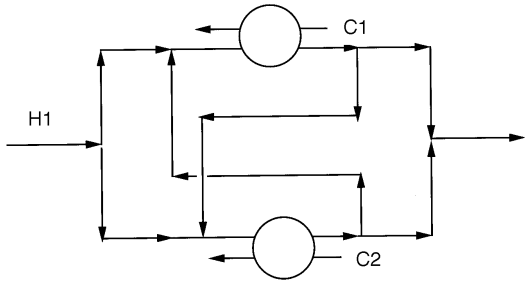


Fully integrated

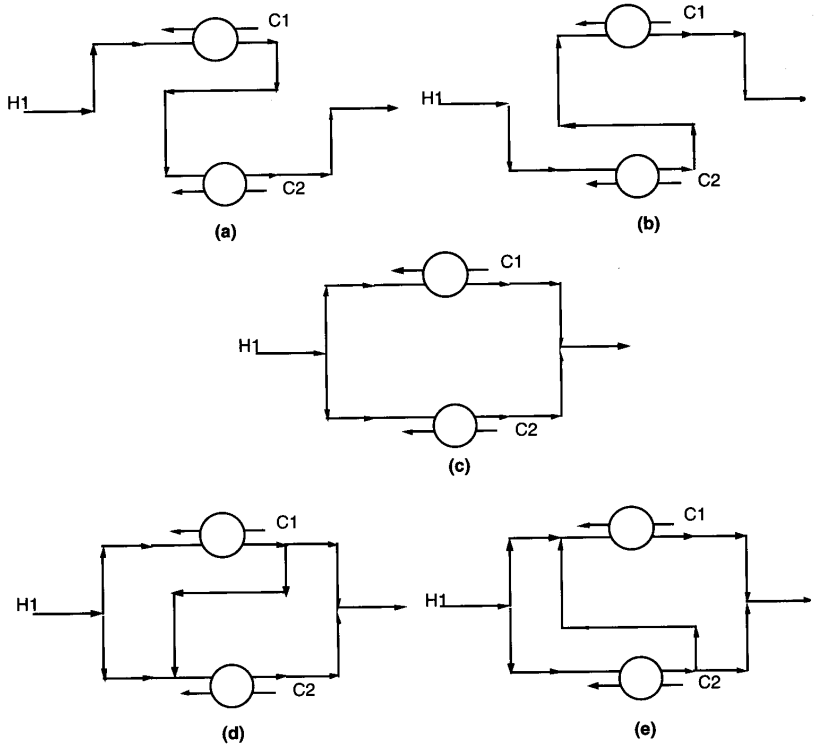
Need to consider both **thermodynamic constraints** and **specific heat needs**!

For 1 CS + 1 HS, only **one** pairing possible.

With 2 CS + 1 HS, **superstructure** is (Biegler et al., 1997):



Alternatives embedded in superstructure:



For more general cases, possible to use a *pairing matrix*, e.g.:

	H1	H2	H3
C1			
C2	•	•	
C3			

0 dots — 1 alternative.
1 dot — 9 alternatives.
2 dots — *vertical* (1 + 2 streams) or *diagonal* (2 + 2 streams).
3 dots — etc.

+ serial / parallel arrangements ⇒ **Thousands of alternatives can be enumerated.**

➪ **Representation** and **solution** efficiencies are important because the selection problems generated are often **NP-hard**.

Current PSE applications (Grossmann and Westerberg, 2000)

Design	Operations
Energy recovery networks	Scheduling
Distillation systems	Multiperiod planning and optimization
Reactor networks	Data reconciliation
Hierarchical decomposition of flowsheets	Real-time optimization
Superstructure optimization	Flexibility analysis
Multiproduct & multipurpose batch plants	Fault diagnosis

Control	Support tools
Model predictive control	Sequential-modular simulation
Controllability	Equation-based simulation
Robust control	A.I. / Expert Systems
Nonlinear control	Large-scale NLP
Statistical process control	Optimization of DAEs
Process Monitoring	Mixed-integer NLP
	Global optimization

➪ Process **improvement** and **diagnosis** are often major driving forces behind **process modeling**.

2. Numerical methods for solution and optimization of process models

Solution methods provide often inspiration for optimization.

2.1. Nonlinear equations

$$f(x) = 0$$
$$x \in \mathbb{R}^n, \; f \text{ nonlinear,} \qquad \text{solution is } f(x^*) = 0$$

Basic algorithm is **Newton's method**. Defining

$$\begin{aligned} J_k &= \nabla f^{\mathrm{T}}(x_k) \in \mathbb{R}^{n \times n} \\ d_{N,k} &= x_{k+1} - x_k \qquad \Rightarrow \qquad \boxed{J_k \cdot d_{N,k} = -f_k} \\ f_k &= f(x_k) \end{aligned} \tag{1}$$

Assume $f(x)$ 2x differentiable, $|J_k| \neq 0$, $\forall x_k$, x_0 **close to solution** (Kantarovich, 1937).

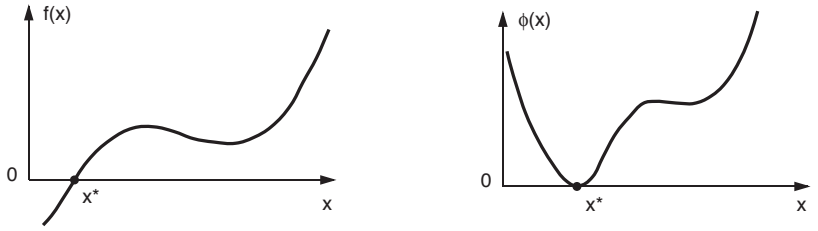
2.1.1. Convergence characteristics

- local **quadratic convergence**:

$$\frac{\|x_{x+1} - x^*\|}{\|x_k - x^*\|^2} \leq K$$

- **global convergence**:
  - ▷ (1) can fail, need to be extended as optimization problem:

$$f(x) = 0 \quad \rightarrow \quad \min_x \phi(x) = f^{\mathrm{T}}(x) f(x)$$



Try to enforce **descent property** for $\phi(x)$:

$$\phi(x_{k+1}) \leq \phi(x_k)$$

Directional derivative:

$$\nabla \phi^{\mathrm{T}}(x_k) \cdot d_{N.k} = -2\phi(x_k) \leq 0$$

➪ **descent direction** guaranteed.

Use **line-search procedure**:

$$x_{k+1} = x_k + \alpha d_{N,k}, \qquad \alpha \in \mathbb{R}.$$

- **Armijo** (1966):
$$\phi(x_k + \alpha d_k) \leq \phi(x_k) + \delta(\nabla\phi^{\mathrm{T}}(x_k) \cdot d_k)\alpha$$
- Also **non-monotonic** line searches (Grippo et al., 1986).

Derivatives J_k can be difficult and expensive to obtain, especially for large problems:

- consider **sparsity** of Jacobian matrix.
- approximate by **finite-differences**:
$$\frac{\partial f_i}{\partial x_j} \simeq \frac{f(\cdots, x_j + \varepsilon_j, \cdots) - f(\cdots, x_j, \cdots)}{\varepsilon_j}$$
  - ▷ $n+1$ function evaluations, $\varepsilon^* \sim \sqrt{E_a}$.

2.1.2. Quasi-Newton methods

Similar to (1):

$$B_k \underbrace{(x_k - x_{k-1})}_{s_k} = \underbrace{(f_k - f_{k-1})}_{y_k} \tag{2}$$

For $n > 1$, (2) not uniquely defined. Popular choice is **Broyden's method**:

$$\begin{aligned} \min_{B_k} \quad & \|B_k - B_{k-1}\|_F \\ \text{s.t.} \quad & B_k s_k = y_k \end{aligned}$$

Solution is recursive update (rank-1):

$$B_k = B_{k-1} - \frac{(y_k - B_{k-1} s_k s_k^{\mathrm{T}})}{s_k^{\mathrm{T}} s_k}$$

Also possible **inverse**, **factored** updates.

Convergence is now just **superlinear** (***local property***).

2.1.3. Trust-region approaches

Instead of line-search, constrain **size** of allowed steps:

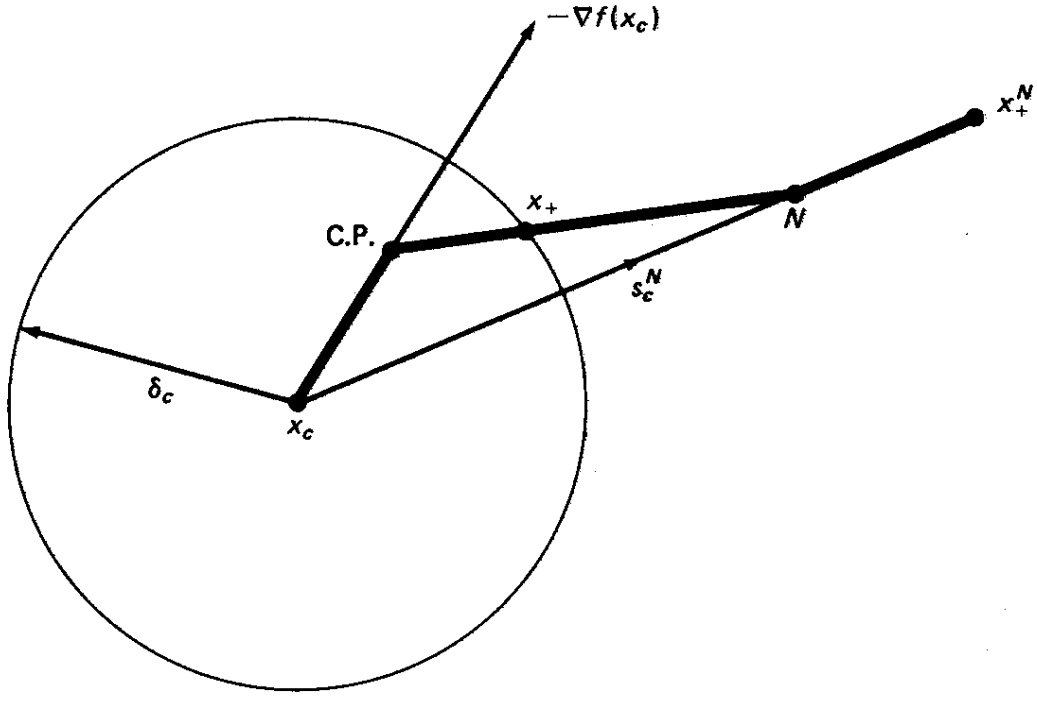
$$\begin{aligned} \min_{d_k} \quad & \phi(x_k) + \nabla\phi^{\mathrm{T}}(x_k) \cdot x_k + \frac{1}{2} d_k^{\mathrm{T}} H(x_k) d_k \\ \text{s.t.} \quad & \|d_k\|_2 \leq \delta_c \end{aligned} \tag{3}$$

For $\delta_c \in [0, +\infty[$, direction is combination of **steepest descent** and **Newton** directions. Therefore possible:

- **direct** (approximate) solution of (3).
- **empirical** solution (Levenberg-Marquardt):

$$(H_k + \lambda I) d_k = -\nabla\phi(x_k)$$

- **double-dogleg** strategy:



(Dennis and Schabel, 1983)

2.1.4. Homotopy (continuation) methods

Instead of tackling $f(x) = 0$, solve a continuous of subproblems, e.g.,

$$h(x, \lambda) = f(x) - (1 - \lambda) f(x_0)$$

$$\lambda = 0 \quad \Rightarrow \quad x^* = x_0, \qquad\qquad \lambda = 1 \quad \Rightarrow \quad h(x, 1) = f(x).$$

Also possible to use dynamic form of models:

$$f(x) = 0 \qquad \rightarrow \qquad \begin{cases} \dfrac{dx}{dt} = f(x) \\ x(0) = x_0 \end{cases}$$

Using, e.g., Euler's method:

$$\frac{x_{k+1} - x_k}{\Delta t} = f(x_k) \qquad \Rightarrow \qquad x_{k+1} = x_k + \Delta t f(x_k).$$

Hence, effect of nonlinearities attenuated when $\Delta t \rightarrow 0$.

- Can also be used to find **multiple solutions**.

2.2. Practicalities

For more difficult cases, reformulation might be convenient:

$$f(x) = 0 \quad \Leftrightarrow \quad \begin{aligned} \min_{x,\varepsilon} \quad & \|\varepsilon\| \\ \text{s.t.} \quad & f(x) = \varepsilon \end{aligned}$$

This **feasible path** approach can also be used to **handle constraints**. For a problem of the form

$$\begin{aligned} f(x) &= 0 \\ l \leq x &\leq u \end{aligned}$$

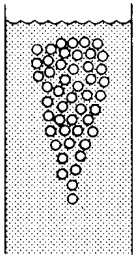
the direct application of variations of Newton's method or trust region can lead to **failures in the model equations**. Instead of (1), consider

$$\begin{aligned} \min_{d_k,\varepsilon} \quad & \|\varepsilon\| \\ \text{s.t.} \quad & f(x_k) + J_k \cdot d_k = \varepsilon \\ & l \leq d_k + x_k \leq u \end{aligned}$$

- Combine with a **line-search** strategy.
- Requires the solution of an **LP** at each iteration (expensive).
- Directions generated are **always feasible**.
- When possible, $\varepsilon = 0 \Rightarrow$ **basic Newton direction**.
- Possible to use l_1, l_2, or l_∞ **norms**. Distinct work involved, convergence properties. In practice, search directions generated are very **similar**.

  Preferred l_1 **norm**. Behavior similar to **mixed complementarity problems** (MCP).
- Extremely useful to **avoid multiple solutions**, e.g., without physical meaning.

Example: Ethanol production with biomass immobilization



Model equations for particles:

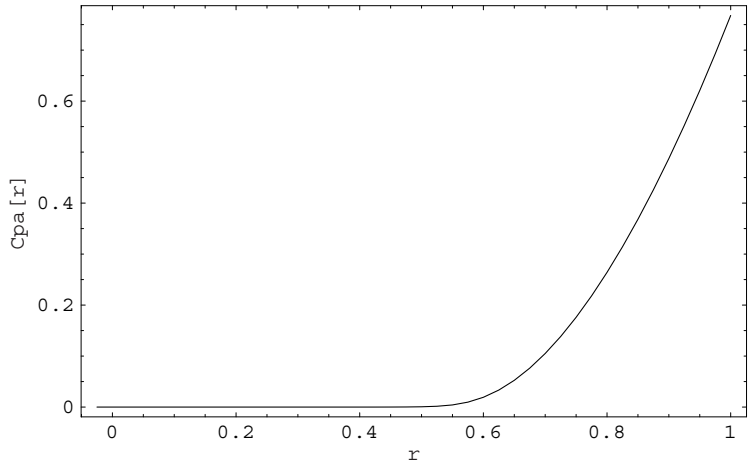
$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial C_p}{\partial r}\right)+\psi(C_a,C_p)=0$$

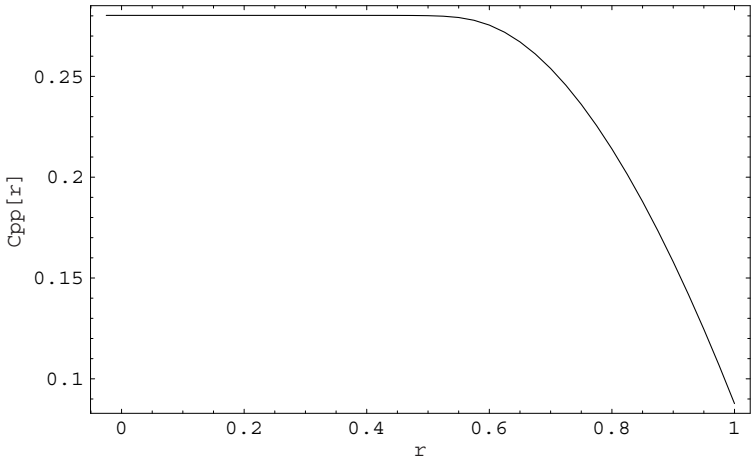
$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial C_a}{\partial r}\right)-\psi(C_a,C_p)=0$$

$$r=0 \quad \Rightarrow \quad \frac{\partial C_a}{\partial r}=\frac{\partial C_a}{\partial r}=0$$

$$r=1 \quad \Rightarrow \quad \frac{\partial C_a}{\partial r}=\mathrm{Bi}_a(1-C_a); \quad \frac{\partial C_a}{\partial r}=\mathrm{Bi}_p(1-C_p)$$

Solution profiles:

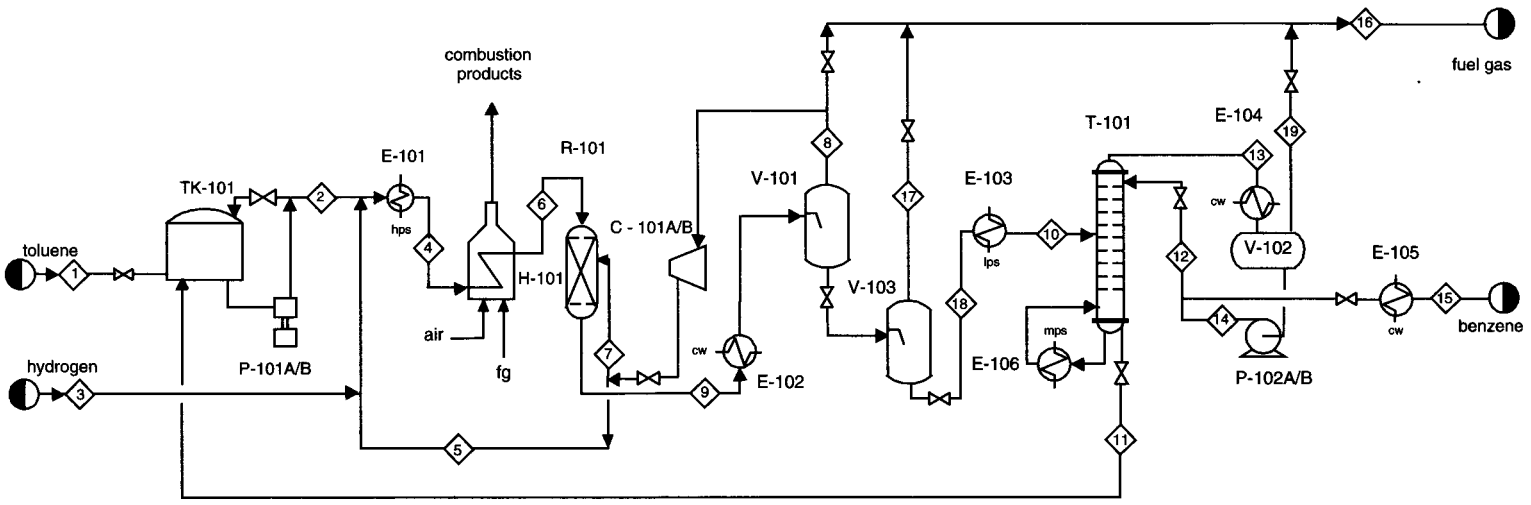




- **Finite differences**, 2nd order centered, 40 points.
- **Many** solutions without physical meaning!

2.3. Application: Process Flowsheeting

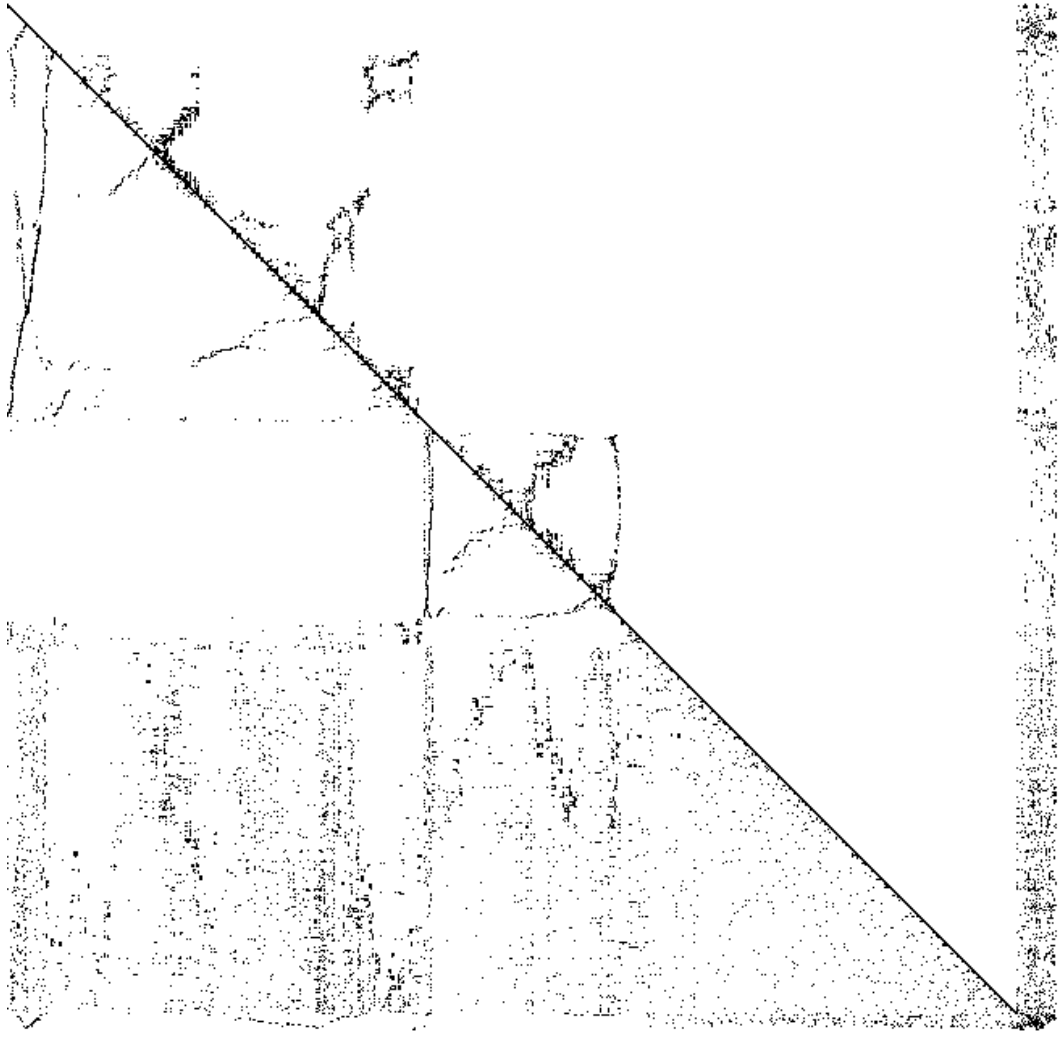
TK-101	P - 101 A/B	E - 101	H - 101	R - 101	C - 101 A/B	E - 102	V - 101	V - 103	E - 103	E - 106	T - 101	E - 104	V - 102	P - 102 A/B	E - 105
Toluene Storage Tank	Toluene Pump	Feed Preheat.	Feed Heater	Reactor	Recycle Gas Compssor	Reactor Effluent Cooler	H.P Phase Separtor	L.P Phase Separtor	Tower Feed Heater	Benz. Reboiler	Benz. Tower	Benz. Conden.	Reflux Drum	Reflux Pumps	Product Cooler



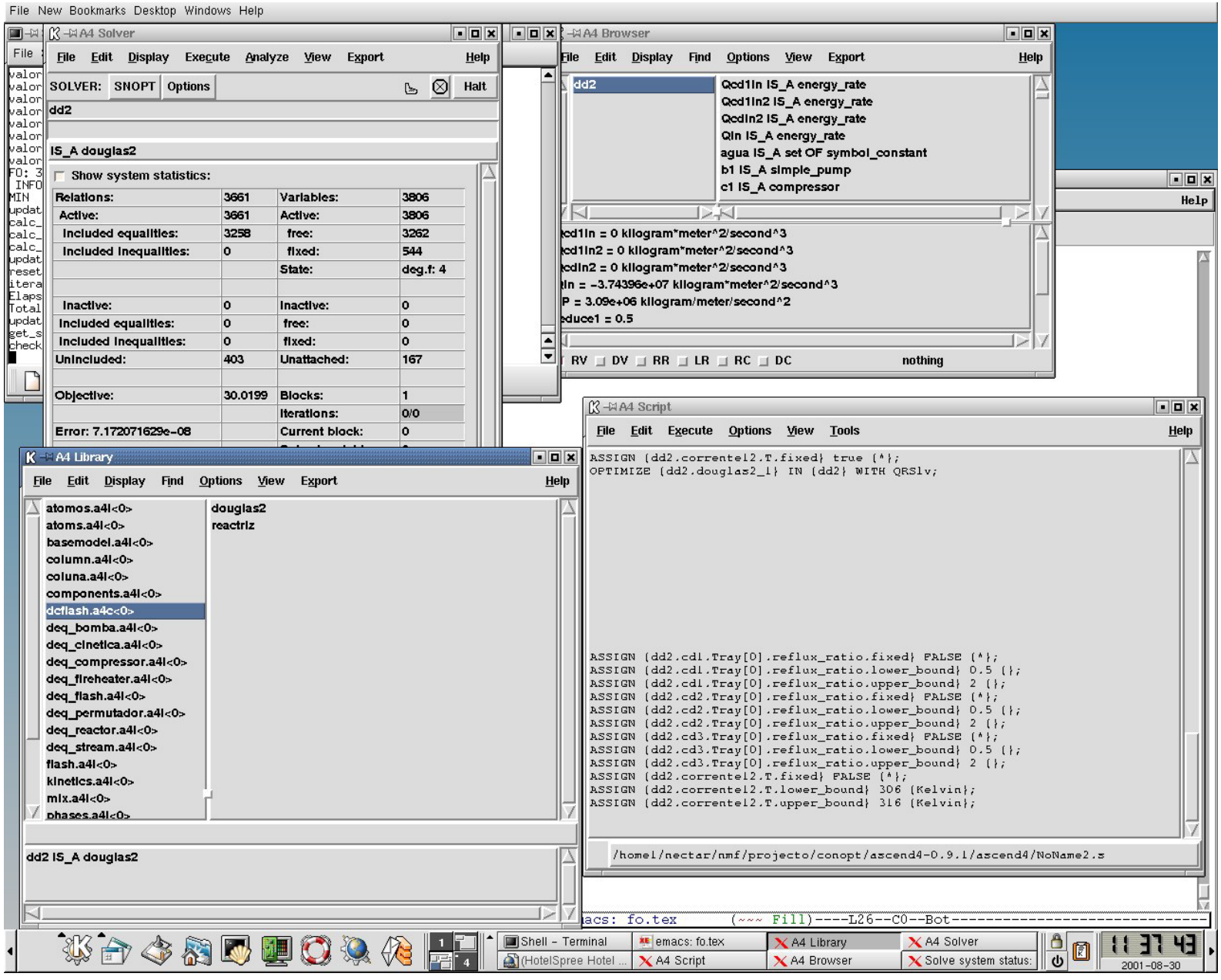
(Turton et al., 1998)

- **Very large** problems (10^4 – 10^5 variables).
- Highly **sparse**.

Typical sparsity diagram:



- **Sequential-modular**:
  - Follow order of material processing.
  - Need to identify **tear streams**. Can be used with simple numerical methods (e.g., successive substitutions).
  - Little flexibility for process specifications.
- **Simultaneous solution**:
  - Identify subsets of equations that require simultaneous solution (**strongly connected** in process digraph).
  - Block partition, at **equation level**. Derivatives only required for diagonal blocks.
  - Very fast, with **good** starting values.
- Based on **split-fractions**:
  - 2-step **successive substitutions**. Very **robust**!
- Require the aid of special simulators:
  - **ASPEN** <http://www.aspentech.com/>.
  - **gPROMS** <http://www.psenterprise.com/>
  - **ASCEND** <http://www.cs.cmu.edu/~ascend/>


File New Bookmarks Desktop Windows Help
A4 Solver
File Edit Display Execute Analyze View Export Help
SOLVER: SNOPT Options Halt
dd2
IS_A douglas2
Show system statistics:
Relations: 3661 Variables: 3806
Active: 3661 Active: 3806
Included equalities: 3258 free: 3262
Included inequalities: 0 fixed: 544
State: deg.f: 4
Inactive: 0 Inactive: 0
Included equalities: 0 free: 0
Included inequalities: 0 fixed: 0
Unincluded: 403 Unattached: 167
Objective: 30.0199 Blocks: 1
Iterations: 0/0
Error: 7.172071629e-08 Current block: 0
A4 Browser
File Edit Display Find Options View Export Help
dd2
Qcd1In IS_A energy_rate
Qcd1In2 IS_A energy_rate
Qcdln2 IS_A energy_rate
Qln IS_A energy_rate
agua IS_A set OF symbol_constant
b1 IS_A simple_pump
c1 IS_A compressor
RV DV RR LR RC DC nothing
A4 Library
File Edit Display Find Options View Export Help
atomos.a4l<0>
atoms.a4l<0>
basemodel.a4l<0>
column.a4l<0>
coluna.a4l<0>
components.a4l<0>
dcflash.a4c<0>
deq_bomba.a4l<0>
deq_cinetica.a4l<0>
deq_compressor.a4l<0>
deq_fireheater.a4l<0>
deq_flash.a4l<0>
deq_permutador.a4l<0>
deq_reactor.a4l<0>
deq_stream.a4l<0>
flash.a4l<0>
kinetics.a4l<0>
mix.a4l<0>
phases.a4l<0>
douglas2
reactriz
dd2 IS_A douglas2
A4 Script
File Edit Execute Options View Tools Help
ASSIGN {dd2.corrente12.T.fixed} true {*};
OPTIMIZE {dd2.douglas2_1} IN {dd2} WITH QRSlv;
ASSIGN {dd2.cd1.Tray[0].reflux_ratio.fixed} FALSE {*};
ASSIGN {dd2.cd1.Tray[0].reflux_ratio.lower_bound} 0.5 {};
ASSIGN {dd2.cd1.Tray[0].reflux_ratio.upper_bound} 2 {};
ASSIGN {dd2.cd2.Tray[0].reflux_ratio.fixed} FALSE {*};
ASSIGN {dd2.cd2.Tray[0].reflux_ratio.lower_bound} 0.5 {};
ASSIGN {dd2.cd2.Tray[0].reflux_ratio.upper_bound} 2 {};
ASSIGN {dd2.cd3.Tray[0].reflux_ratio.fixed} FALSE {*};
ASSIGN {dd2.cd3.Tray[0].reflux_ratio.lower_bound} 0.5 {};
ASSIGN {dd2.cd3.Tray[0].reflux_ratio.upper_bound} 2 {};
ASSIGN {dd2.corrente12.T.fixed} FALSE {*};
ASSIGN {dd2.corrente12.T.lower_bound} 306 {Kelvin};
ASSIGN {dd2.corrente12.T.upper_bound} 316 {Kelvin};
/home1/nectar/nmf/projecto/conopt/ascend4-0.9.1/ascend4/NoName2.s
Shell - Terminal
emacs: fo.tex
A4 Library
A4 Solver
(HotelSpree Hotel ...
A4 Script
A4 Browser
Solve system status:
2001-08-30

3. Nonlinear optimization

$$\begin{aligned} \min_{x} \quad & \phi(x) \\ \text{s.t.} \quad & h(x) = 0 \\ & g(x) \leq 0 \\ & x \in \mathbb{R}^n \end{aligned} \tag{4}$$

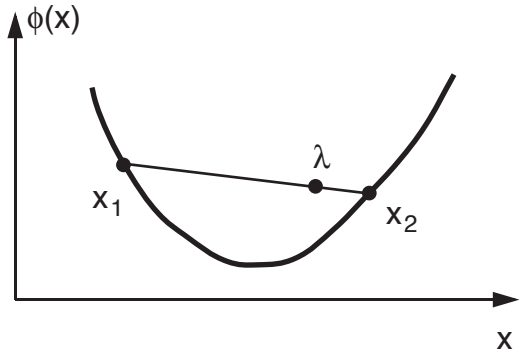
Important problem characteristics:

- **Linear / nonlinear:**

	Objective	Constraints
LP	L	L
QP	Q	L
NLP	NL	NL

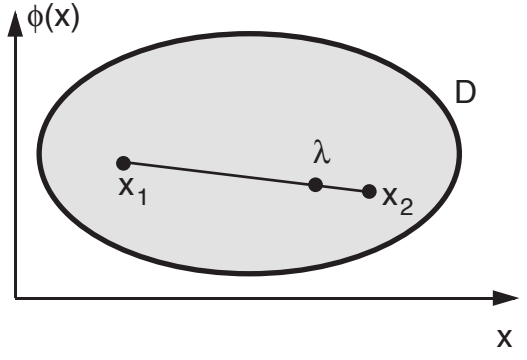
- **Convexity:**
  - ▷ **Functions:**

$$\phi\left[\lambda x_1 + (1-\lambda)x_2\right] \leq$$
$$\leq \lambda\phi(x_1) + (1-\lambda)\phi(x_2), \quad \lambda \in [0,1].$$



  - ▷ **Domains:**

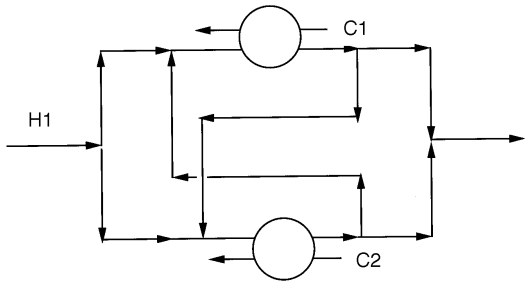
$$z = \lambda x_1 + (1-\lambda)x_2, \quad \lambda \in [0,1].$$



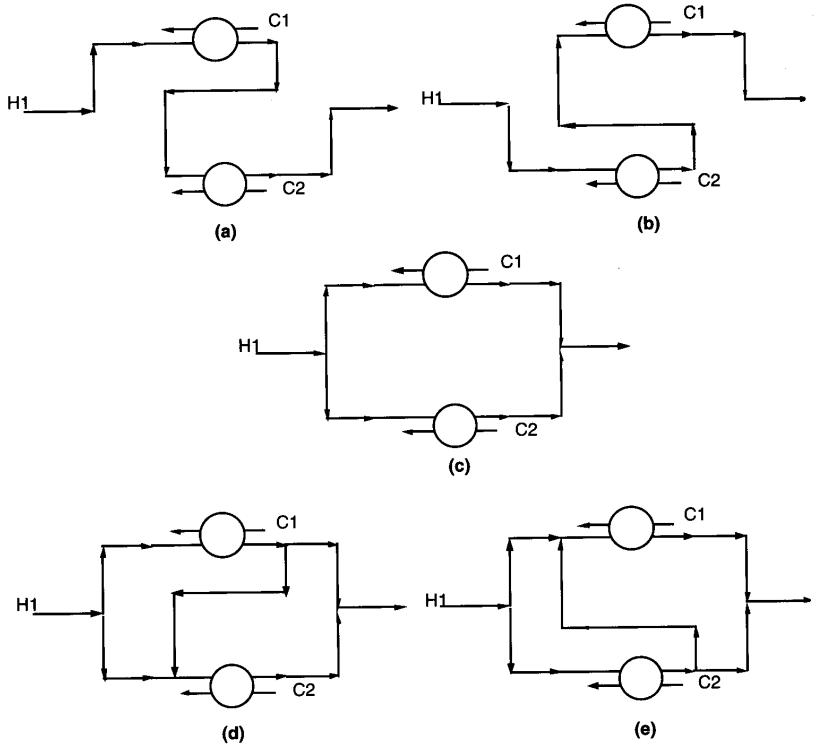
If $\phi(x)$ is **convex**, $h(x)$ **linear**, $g(x)$ **convex**, a **local solution** of (4) is also a **global solution**.

Implications for process design:

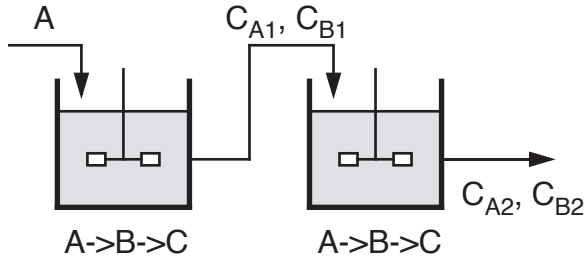
Superstructure for 2 CS + 1 HS:



Need to optimize separately **each** alternative:



Example: Reactor network design (Ryoo and Sahinidis, 1995)



Design system that **maximizes** C_B at output, assuming **isothermal** operation:

$$
\begin{aligned}
& \min_{C_i, V_i} \quad -C_{B2} \\
\text{s.t.} \quad & C_{A1} + k_1 C_{A1} V_1 = 1 \\
& C_{A2} - C_{A1} + k_1 k_2 C_{A2} V_2 = 0 \\
& C_{B1} + C_{A1} + k_3 C_{B1} V_1 = 1 \\
& C_{B2} - C_{B1} + C_{A2} - C_{A1} + k_3 k_4 C_{B2} V_2 = 0 \\
& V_1^{0.5} + V_2^{0.5} \leq 4 \\
& 0 \leq C_i \leq 1 \\
& 0 \leq V_i \leq 16
\end{aligned}
$$

- Problem has **3 local solutions**, with close objective values.
- **Global** solution:

$C_{A1} = 0.772$	$C_{B1} = 0.204$	$V_1 = 3.04$
$C_{A2} = 0.517$	$C_{B2} = 0.388$	$V_2 = 5.10$

3.1. Optimality conditions

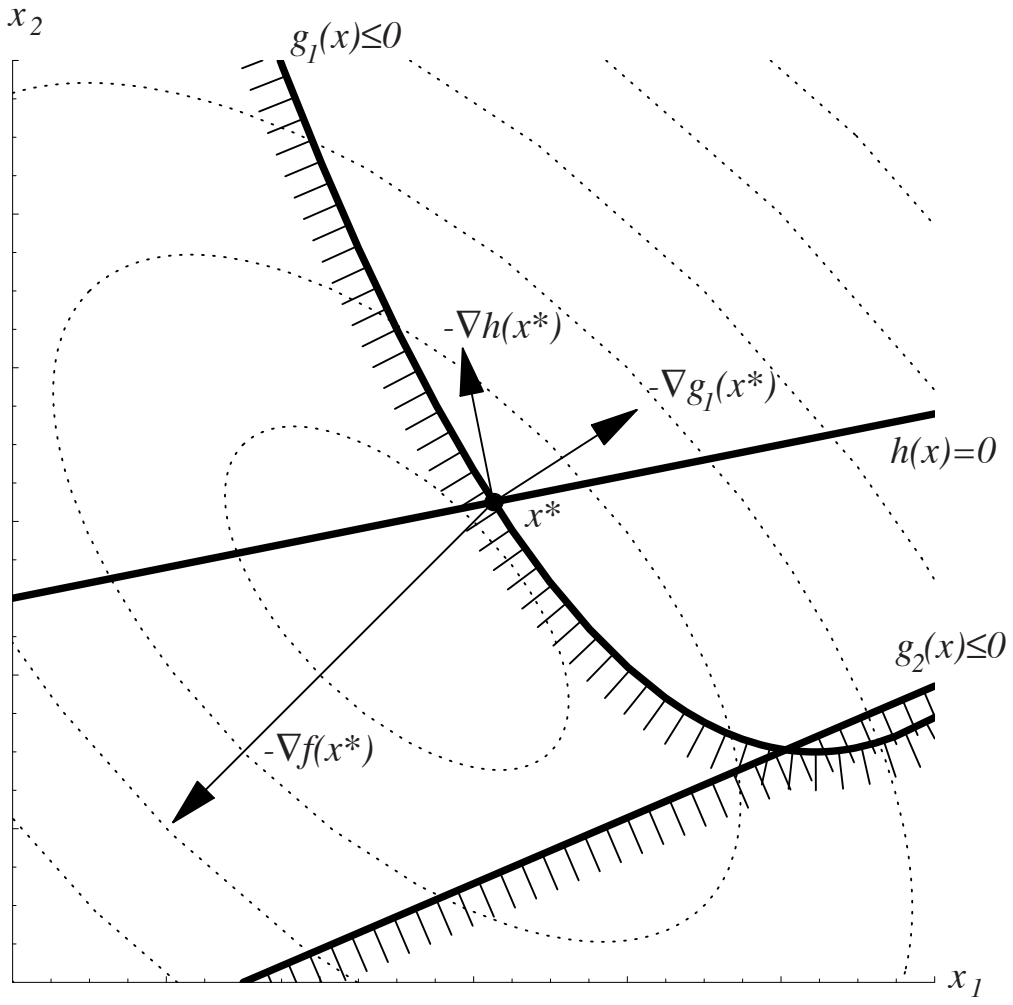
$$\mathscr{L}(x,\lambda,\mu) = \phi(x) + \lambda^{\mathrm{T}} h(x) + \mu^{\mathrm{T}} g(x)$$

$$\boxed{\nabla_x \mathscr{L} = 0} \quad \Rightarrow \quad \nabla\phi(x^*) + \nabla h(x^*)\cdot\lambda + \nabla g(x^*)\cdot\mu = 0 \qquad (5)$$

$$\boxed{\nabla_\lambda \mathscr{L} = 0} \quad \Rightarrow \quad h(x) = 0$$

$$\text{complementarity} \quad \Rightarrow \quad \begin{array}{l} \mu_i g_i(x^*) = 0 \\ \mu_i \geq 0 \end{array}$$

(5) can be interpreted as a **force balance** for equilibrium:



3.2. Successive Quadratic Programming (SQP)

K.T. conditions of (5) are **system of NL equations** $\Rightarrow$ iterative solution. Consider:

$$\begin{array}{ll} \min\limits_{x} & \phi(x) \\ \text{s.t.} & c(x) = 0 \end{array} \qquad \Rightarrow \qquad \mathscr{L}(x,\lambda) = \phi(x) + \lambda^{\mathrm{T}} c(x)$$

K.T. conditions:

$$\begin{aligned} \nabla\phi(x) + \nabla c(x)\lambda &= 0 \\ c(x) &= 0 \end{aligned}$$

First order expansion around (x_k, λ_k):

$$\begin{bmatrix} \nabla\phi(x_k) + \nabla c(x_k)\lambda_k \\ c(x_k) \end{bmatrix} + \begin{bmatrix} \nabla^2\phi(x_k) + \nabla^2 c(x_k)\lambda_k & \nabla c(x_k) \\ \nabla c^{\mathrm{T}}(x_k) & 0 \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta\lambda \end{bmatrix} = 0$$

These are equivalent to the K.T. conditions of

$$\begin{aligned} \min_{\Delta x_k} \quad & \phi(x_k) + \nabla\phi^{\mathrm{T}}(x_k)\Delta x_k + \frac{1}{2}\Delta x_k^{\mathrm{T}}\left[\nabla^2\phi(x_k) + \nabla^2 c(x_k)\lambda_k\right]\Delta x_k \\ \text{s.t.} \quad & c(x_k) + \nabla c(x_k)^{\mathrm{T}}\Delta x = 0 \end{aligned}$$

- QP solution at each iteration for **search direction**.
- Need 2nd order derivatives for both **objective** and **constraints**.
- Can also use **quasi-Newton approximation**:

$$B_k = B_{k-1} - \frac{B_{k-1}s_k s_k^{\mathrm{T}} B_{k-1}}{s_k^{\mathrm{T}} B_{k-1} x_k} + \frac{y_k y_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k}$$ **BFGS (rank-2 update)**

$$s_k = x_{k+1} - x_k$$

$$y_k = \nabla\mathscr{L}(x_{k+1}, \lambda_{k+1}) - \nabla\mathscr{L}(x_k, \lambda_k)$$

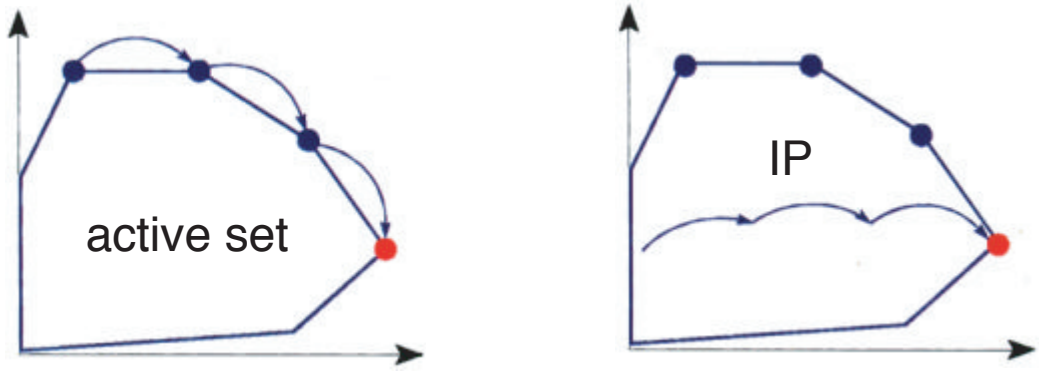
3.3. Interior-point methods

Complementarity conditions of K.T. equations can be difficult to solve (**combinatorial**, **ill-conditioned**).

Barrier methods

$$\begin{aligned} \min_x \quad & \phi(x_k) - \mu \sum_i \ln s_i \\ \text{s.t.} \quad & h(x_k) = 0 \\ & g(x_k) + s = 0 \end{aligned}$$

Solve **sequence of problems**, where $\mu \to 0$ (**homotopy parameter**).



Tradeoff: more iterations (several μ) vs. **more expensive QPs**.

3.4. Reduced gradient methods (GRG)

Similar to **simplex methods** for LP.

- Decide which constraints are **active / inactive**.
- Partition variables as **dependent / independent**.
- Solve **unconstrained problem** in the space of **independent variables**.

Consider:

$$\begin{array}{ll} \min\limits_{z} & \phi(x) \\ \text{s.t.} & h(z) = 0 \\ & l \le z \le u \end{array} \qquad \text{+ slack variables.}$$

Partition variables as:

$$z = \begin{bmatrix} z_I \\ \hline z_D \end{bmatrix} \qquad \Rightarrow \qquad h(z_I, z_D) = 0$$

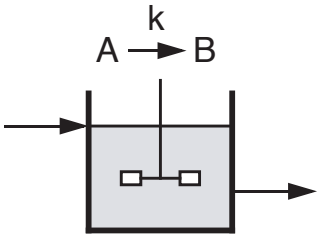
Constrained derivative:

$$\frac{d\phi}{dz_I} = \frac{\partial \phi}{\partial z_I} + \frac{dz_D}{dz_I}\frac{\partial \phi}{\partial z_D}$$

$$\frac{dz_D}{dz_I} = [\nabla_{z_I} h] \cdot [\nabla_{z_D} h]^{-1} \qquad \textbf{implicit form}$$

- Always takes **feasible steps**.
- Very **robust implementations** available (e.g., CONOPT3).
  - ▷ practical details (**initialization**, **scaling**, **sparsity**).
  - ▷ combined with **other solution techniques** (Newton steps, SQP).

Example: Kinetic parameter estimation (CSTR)



$$\min_{\theta,z} \quad \sum_{j=1}^{r}\sum_{i=1}^{m} \frac{(z_{ij} - z_{ij,\mathrm{mod}})^2}{\sigma_i^2}$$

$$\text{s.t.} \quad \frac{(z_1 - z_2)}{\tau} - kz_2 = 0$$

$$-\frac{z_3}{\tau} + kz_2 = 0$$

$$\frac{z_4 - z_5}{\tau} + \frac{(-\Delta H)}{\rho C_p} kz_2 = 0$$

$$k = \theta_1 \exp\left(-\frac{\theta_2}{z_5}\right)$$

- For **batch** data, need optimization of a DAE model.
- Can easily be solved within GAMS or AMPL — **local** solutions.

4. Discrete optimization

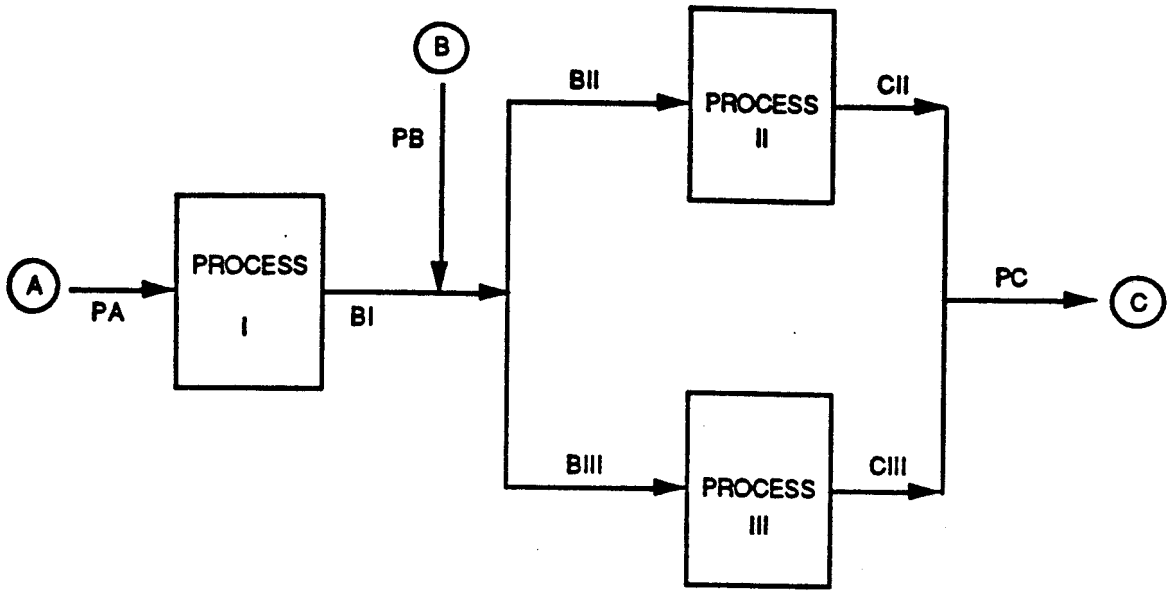
$$\begin{aligned} \min_{x,y} \quad & \phi(x,y) \\ \text{s.t.} \quad & h(x,y) = 0 \\ & g(x,y) \leq 0 \\ & x \in \mathbb{R}^n,\ y \in \mathbb{N}^m \end{aligned}$$

Possible **integer** variables ($y \in \mathbb{N}$) or **binary** variables ($y \in \{0,1\}$). Can assume **only binary** present. Some algorithms take advantage of both, others require just binary (e.g., like simple bounds).

Problem classes:

Type	Objective	Constraints	Variables
IP	L	L	I
LP	L	L	R
MILP	L	L	I, R
NLP	NL	NL	R
MINLP	NL	NL	I, R
NIP	NL	NL	I

Process example: Design of a chemical complex (Sahinidis and Grossmann, 1991)



Decisions:

- C produced, and how much?
- Which alternatives II or III (exclusive)?
- How to obtain B?

Process data:

Process	Conversion (%)	Fixed cost	Variable cost
I	90	1000	250
II	82	1500	400
III	95	2000	550

Species	Prices	Constraints
A	500	≤ 16
B	950	
C	1800	≤ 10

Solution:

$$y_i = \begin{cases} 1, & \text{process } i \text{ selected.} \\ 0, & \text{if not.} \end{cases}$$

Exclusivity of II and III:

$$y_2 + y_3 \leq 1$$

Mass balances:

$$B_1 + P_B = B_2 + B_3 \qquad B_1 = 0.9P_A$$

$$C_2 + C_3 = S_C \qquad C_2 = 0.82B_2$$

$$C_3 = 0.95B_3$$

Constraints:

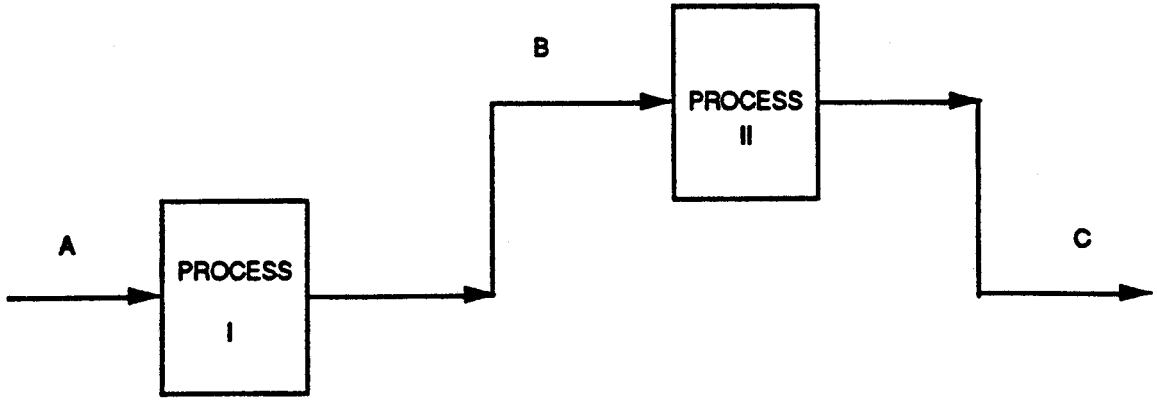
$$P_A \leq 16y_1 \qquad B_3 \leq (10/0.95)y_3$$
$$B_2 \leq (10/0.82)y_2 \qquad S_C \leq 10$$

Objective:

$$\begin{aligned}\phi(\text{Profit}) = -(&1000y_1 + 250P_A + \quad \rightarrow \textbf{ investment}\\ &1500y_2 + 400B_2 +\\ &2000y_3 + 550B_3)\\ &-500P_A - 950P_B \quad \rightarrow \textbf{ purchases}\\ &+1800S_C \quad \rightarrow \textbf{ sales}\end{aligned}$$

➪ **MILP problem.**

Optimal solution:

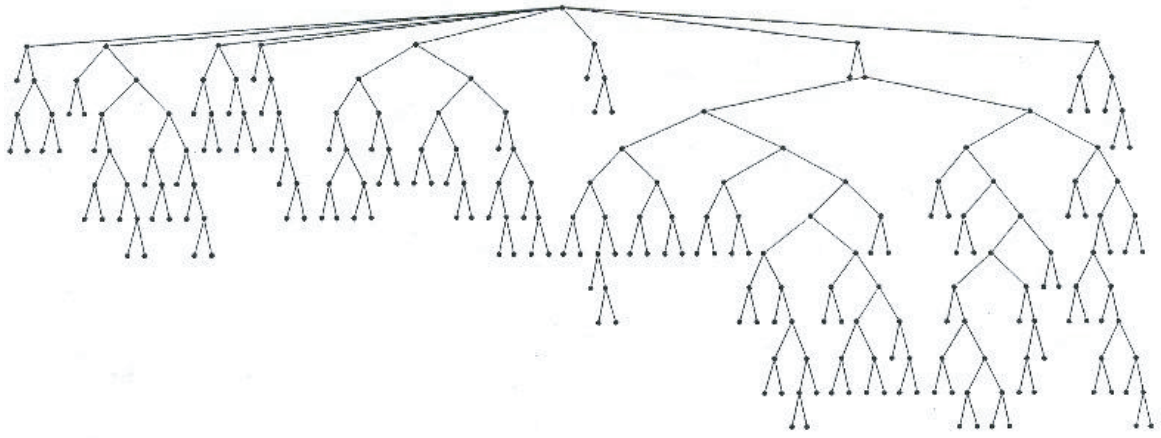


$$\phi = 459.3 \qquad B_2 = 12.195$$
$$P_A = 13.55 \qquad S_C = 10$$

Solution much more complex if formulation was **MINLP**.

4.1. Solution methods for MILP

- **Real relaxation** (LP) + solution rounding.
  - ▷ Dangerous with binary variables.
- **Explicit enumeration**: Build complete **decision tree**:



  - ▷ Only possible with very small problems (brute force).

> For m discrete variables, with n_i distinct values, need to solve $\prod_{i=1}^{m} n_i$ **continuous optimization problems, in** $(n_v - m)$ **variables**.

  - ▷ Several strategies to visit nodes (**depth-first**, **breadth-first**, specific **A.I. heuristics**).

- Method of **cutting planes**:
  - ▷ Start with LP relaxation.
  - ▷ If not integer solution already, **add constraints** to previous problems, solve new LP.
  - ▷ Incremental characterization of **convex hull** of problem.
- **Boolean methods** (only binary variables):
  - ▷ Exploit similarities with **propositional logic** and **theorem proving** (e.g., PROLOG).
- **Branch & Bound:** (implicit enumeration)
  - ▷ Create decision tree (**branch**). Use solution bounds to reduce search space (**bound**).

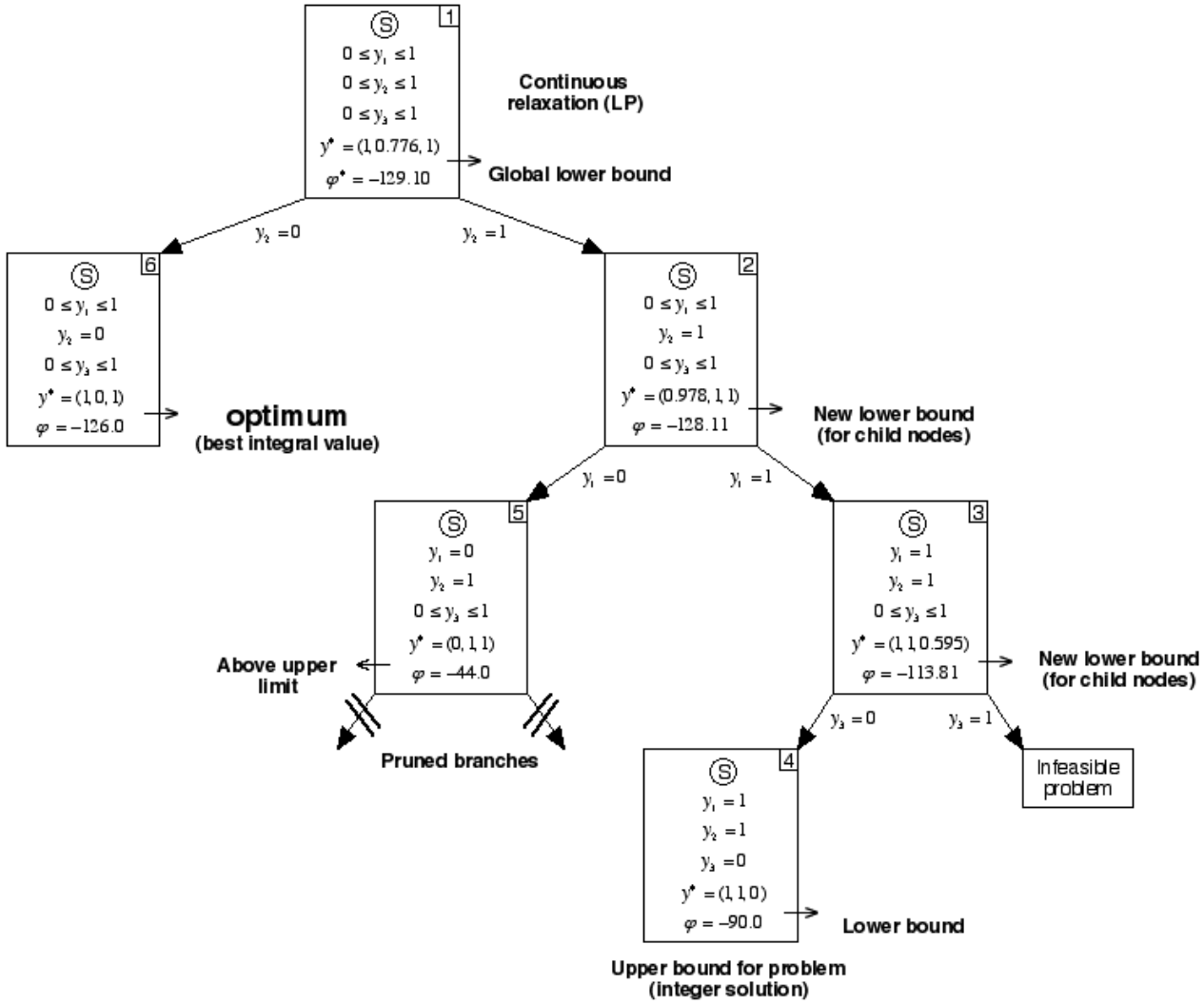
Reference formulation:

$$\begin{aligned} \min_{x,y} \quad & a^{\mathrm{T}} y + c^{\mathrm{T}} x \\ \text{s.t.} \quad & By + Ax \leq b \\ & x \geq 0,\ y \in \{0,1\} \end{aligned}$$

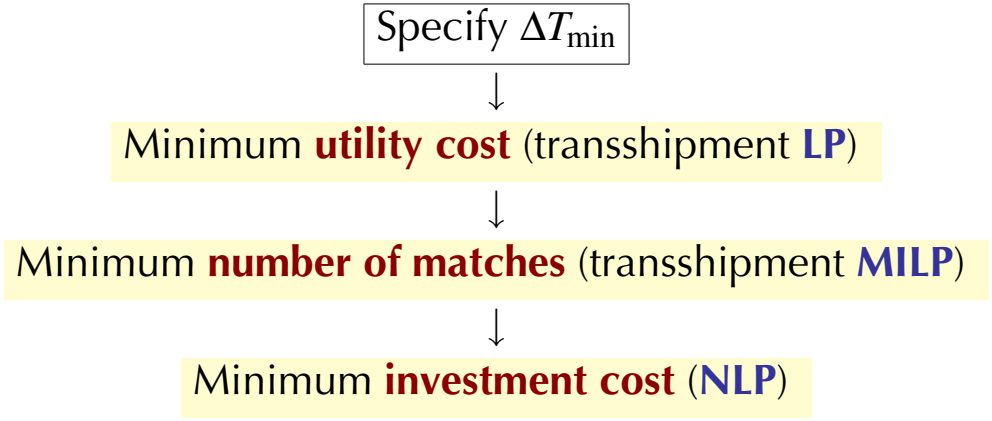
Lower bound: solution of **relaxed LP**.
Upper bound: best **integer solution** so far.

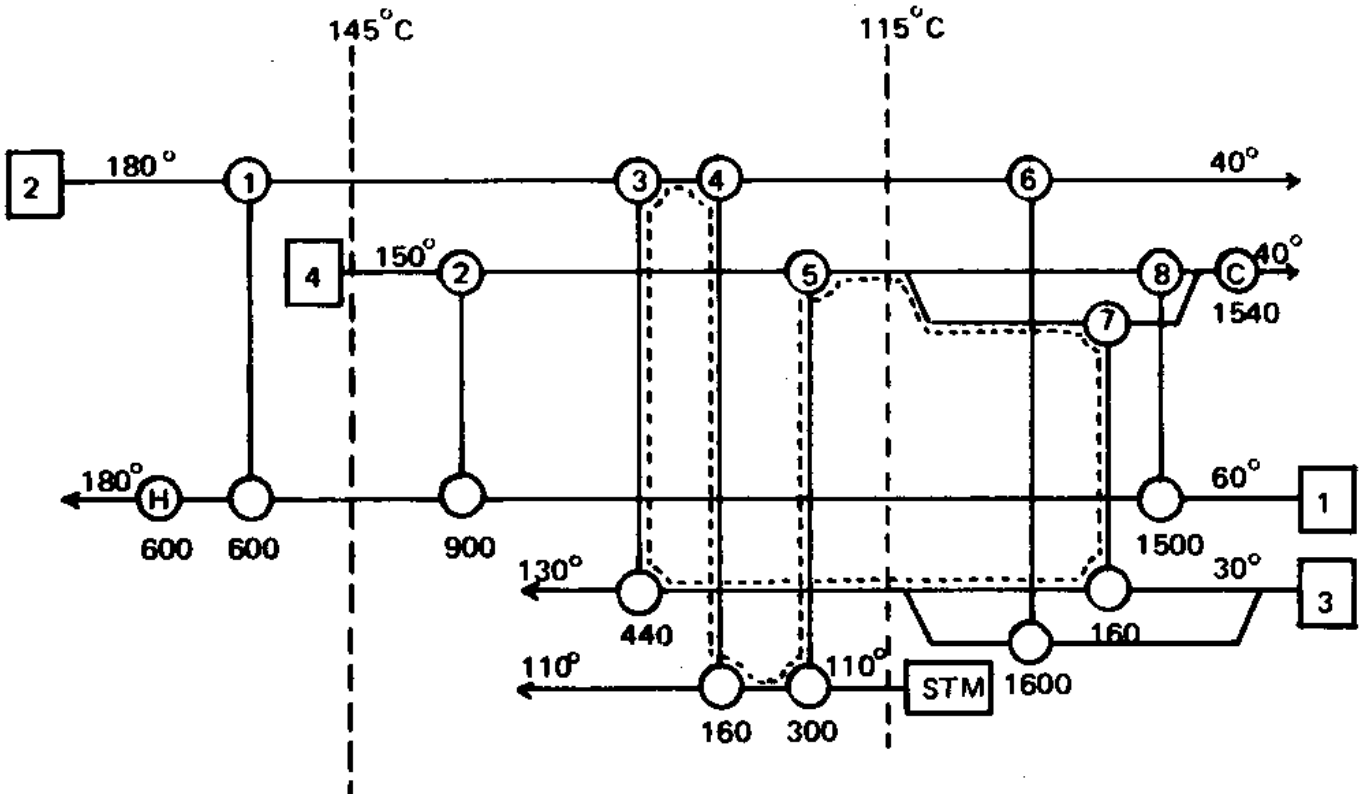
Simple example:

$$\min_{y} \ \varphi = -86y_1 - 4y_2 - 40y_3$$
$$\text{s.a.} \quad 774y_1 + 76y_2 + 42y_3 \le 875 \quad \text{(S)}$$
$$67y_1 + 27y_2 + 53y_3 \le 875$$
$$y_1, y_2, y_3 \in \{0,1\}$$



Example: Process heat integration (targeting)





(Linnhoff et al., 1982)

4.2. Solution of MINLPs

$$\begin{aligned} \min_{x,y}(\max) \quad & \phi(x) + c^{\mathrm{T}} y \\ \text{s.t.} \quad & g(x) + Hy(\bullet)b \qquad\qquad (\bullet) = \{\leq, =, \geq\} \\ & l \leq x \leq u,\ y \in \{0,1\} \end{aligned}$$

NOTE: If $f(y)$ nonlinear, can replace:

$$\begin{aligned} f(y) \rightarrow f(y'), \qquad & y' \in \mathbb{R} \\ y = y' \qquad\qquad & 0 \leq y' \leq 1 \end{aligned}$$

Algorithms for MINLPs (Grossmann, 2001):

- **Generalized Benders decomposition**.
- **Outer approximation** (OA).
- **Extended Cutting Planes**.
- **Simple Branch & Bound** (SBB) (Bussieck and Drud, 2001).

4.3. Modeling with discrete variables

Simple cases:

- **Multiple choice:**
  - ▷ Select **only 1** item: $\sum_i y_i = 1$
  - ▷ Select **at most 1** item: $\sum_i y_i \leq 1$
- If $y = 0$, associated x or $f(x)$ must be 0:

$$f(x) - My \leq 0, \qquad f(x) \in [0, M]$$

- **Implications** ($K \Rightarrow J$):

$$y_k - y_j \leq 0$$

Systematic incorporation of logical information (Ramesh and Grossmann, 1991)

Simple logic relations:

Table 1. Representation of logical relations with linear inequalities

Logical relation	Comments	Logical expression	Representation as linear inequalities
Logical "OR"		$P_1 \vee P_2 \vee \cdots \vee P_r$	$y_1 + y_2 + \cdots + y_r \geq 1$
Logical "AND"		$P_1 \wedge P_2 \wedge \cdots \wedge P_r$	$y_1 \geq 1; y_2 \geq 1; \ldots; y_r \geq 1$
Implication	$P_1 \Rightarrow P_2$ is logically equivalent to $\neg P_1 \vee P_2$	$\neg P_1 \vee P_2$	$1 - y_1 + y_2 \geq 1$ or $y_1 - y_2 \leq 0$
Equivalence	P_1 if and only if P_2 $(P_1 \Rightarrow P_2) \wedge (P_2 \Rightarrow P_1)$	$(\neg P_1 \vee P_2) \wedge (\neg P_2 \vee P_1)$	$y_1 - y_2 \leq 0; y_2 - y_1 \leq 0$ or $y_1 = y_2$
Exclusive "OR" (EOR)	Exactly one of the variables is true	$P_1 \oplus P_2 \oplus \cdots \oplus P_r$	$y_1 + y_2 + \cdots + y_r = 1$
Classification	$Q = \{P_1, P_2, \ldots, P_r\}$ Q is true if any of the variables inside the brackets are true		$y_q = y_1 + \cdots + y_r$

Combine with reduction to **normal conjunctive form** using **logic equivalences**, **DeMorgan laws**, and **distributive properties**.

Example: If product A or B (or both) produced, at least one of the products C,D, or E must also be manufactured.

Let $P_i \equiv$ product i manufactured. Associate $y_i \in \{0,1\}$ with P_i. Equivalent proposition is

$$(P_A \vee P_B) \Rightarrow (P_C \vee P_D \vee P_E)$$

Simplification:

$$(P_A \vee P_B) \Rightarrow (P_C \vee P_D \vee P_E) \quad \rightarrow \quad \overline{(P_A \vee P_B)} \vee (P_C \vee P_D \vee P_E) \quad \rightarrow$$

$$\rightarrow \quad (\bar{P}_A \wedge \bar{P}_B) \vee (P_C \vee P_D \vee P_E) \quad \rightarrow$$

$$\rightarrow \quad [\bar{P}_A \vee (P_C \vee P_D \vee P_E)] \wedge [\bar{P}_B \vee (P_C \vee P_D \vee P_E)]$$

Using the previous table, originates:

$$(1 - y_A) + y_C + y_D + y_E \geq 1$$
$$(1 - y_B) + y_C + y_D + y_E \geq 1$$

Other uses:

- This approach can also be used to model **heuristics**:

$$A \Rightarrow B \quad \rightarrow \quad \begin{array}{ll} y_A - y_B \leq 0 & \text{(logic)} \\ y_A - y_B \leq s & \text{(heuristic)} \end{array}$$

  Penalty can be associated with constraint violations.

- Discrete variables also useful to approximate **non-convex regions** (Williams, 1993) or **functions**. Typical cost functions in ChemE:

$$C = av^b, \qquad b \sim 2/3 \Rightarrow \textbf{nonconvex}$$

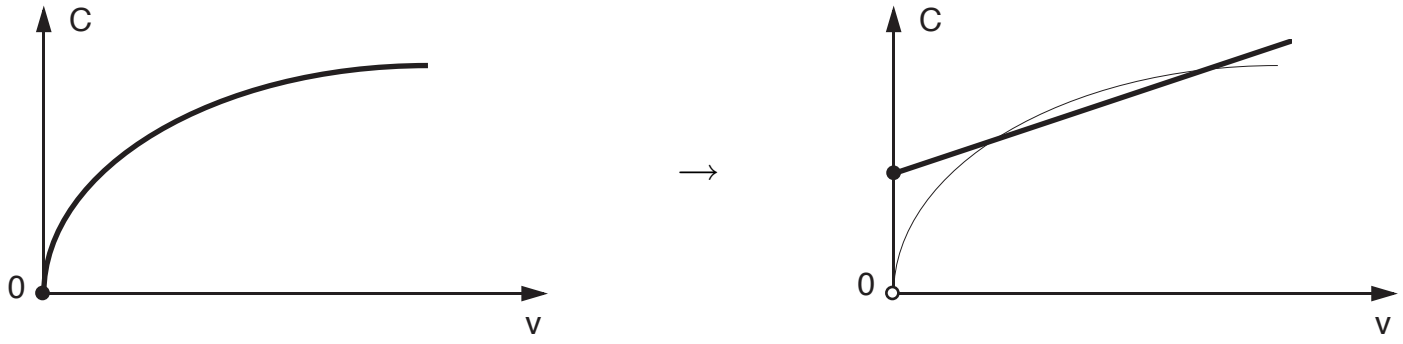
Table 6-4 Typical exponents for equipment cost as a function of capacity

Equipment	Size range	Exponent
Blender, double cone rotary, carbon steel (c.s.)	1.4–7.1 m^3 (50–250 ft^3)	0.49
Blower, centrifugal	0.5–4.7 m^3/s (10^3–10^4 ft^3/min)	0.59
Centrifuge, solid bowl, c.s.	7.5–75 kW (10–10^2 hp) drive	0.67
Crystallizer, vacuum batch, c.s.	15–200 m^3 (500–7000 ft^3)	0.37
Compressor, reciprocating, air-cooled, two-stage, 1035-kPa discharge	0.005–0.19 m^3 (10–400 ft^3/min)	0.69
Compressor, rotary, single-stage, sliding vane, 1035-kPa discharge	0.05–0.5 m^3/s (10^2–10^3 ft^3/min)	0.79
Dryer, drum, single vacuum	1–10 m^2 (10–10^2 ft^2)	0.76
Dryer, drum, single atmospheric	1–10 m^2 (10–10^2 ft^2)	0.40
Evaporator (installed), horizontal tank	10–1000 m^2 (10^2–10^4 ft^2)	0.54
Fan, centrifugal	0.5–5 m^3/s (10^3–10^4 ft^3/min)	0.44
Fan, centrifugal	10–35 m^3/s (2×10^4–7×10^4 ft^3/min)	1.17
Heat exchanger, shell-and-tube, floating head, c.s.	10–40 m^2 (100–400 ft^2)	0.60
Heat exchanger, shell-and-tube, fixed sheet, c.s.	10–40 m^2 (100–400 ft^2)	0.44
Kettle, cast-iron, jacketed	1–3 m^3 (250–800 gal)	0.27
Kettle, glass-lined, jacketed	0.8–3 m^3 (200–800 gal)	0.31

Using discrete variables, the cost functions can be approximated as:

$$C = \begin{cases} 0, & v = 0 \\ C_f + C_v v, & v > 0 \end{cases} \qquad \text{with:}$$

$$y = 0 \;\Rightarrow\; v = 0 \quad \rightarrow \quad \begin{cases} v - M_v y \le 0 \\ v \ge 0 \end{cases}$$

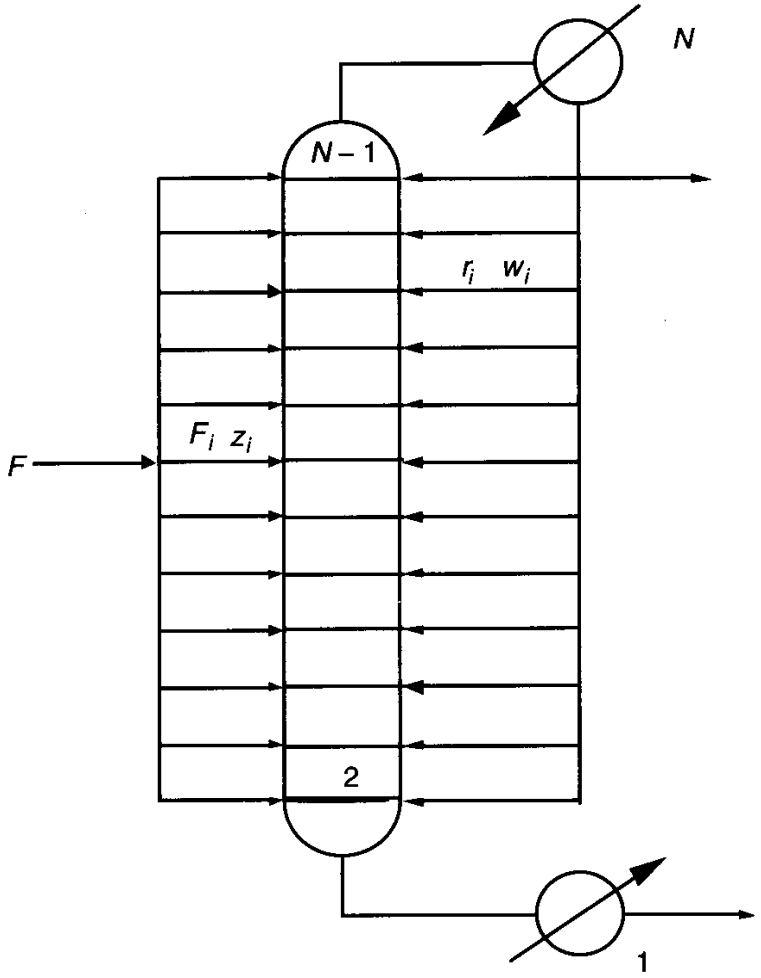


Can also be generalized to approximation using **several linear segments**, or **discontinuous changes** in costs.

Examples:

- Distillation design.
- Flowsheet optimization with detailed models (**fixed structure**).
- Process design.
- Trim loss minimization.

Distillation design:



5. Optimization of systems described by differential equations

In general, more detailed and accurate models in ChemE involve differential equations (e.g., PDE) $\Rightarrow$ need to **solve** and **optimize**.

5.1. Problem types

- **Optimal control:**

$$\begin{aligned} \min_{u(t)} \quad & \phi(z(t_F)) \\ \text{s.t.} \quad & \dot{z} = f(z,u,\theta),\ z(0) = z_0 \\ & h(z,u,\theta) = 0 \\ & g(z,u,\theta) \leq 0 \end{aligned}$$

  - $u(t) \in \mathbb{R}^n$ infinite dimensional $\rightarrow$ **calculus of variations**.
  - E.g., **process control**, **discontinuous processes**.

- **State or parameter estimation:**

$$\begin{aligned} \min_{\theta} \quad & \sum_i [y_{\text{obs}}(t_i) - y_{\text{mod}}(t_i, \theta)]^2 \\ \text{s.t.} \quad & \text{process model} \\ & \text{constraints} \end{aligned}$$

  - ▷ Now finite no. of DOF. Hence can be solved as conventional NLP, if DAE model treated as an entire **I/O block** (**black-box approach**).

- **Design problems:**

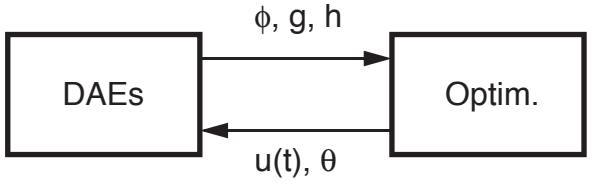
$$\begin{aligned} \min_{v} \quad & \text{(cost, -performance)} \\ \text{s.t.} \quad & \text{process model} \\ & \text{constraints} \end{aligned}$$

  - ▷ E.g., **optimize catalyst**, **nonhomogeneous reactor**, etc.

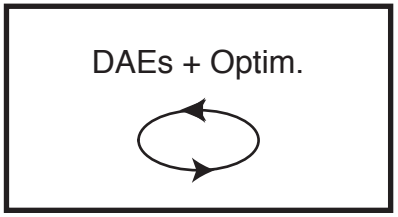
Can involve ODEs or PDEs; here just **ODEs**.

5.2. Strategies for optimization of DAE models

- Direct solution of **optimality conditions** (**Euler-Lagrange**) $\rightarrow$ **variational methods**.
- Analogy with **solution methods**; these usually involve a **discretization** of the ODEs:
  - ▷ **Partial discretization**: only input profile $u(t)$ parametrized. Since the derivatives kept, the optimization and solution are done **sequentially**:



  - ▷ **Full discretization**: input $u(t)$ and state $z(t)$ profiles discretized $\rightarrow$ scalar (large-scale) NLP, for **simultaneous** solution and optimization.



- Also possible **dynamic programming**, e.g. after partial discretization.

5.2.1. Optimality conditions for ODE systems

Using Mayer form:

$$
\begin{aligned}
\min_{u(t)} \quad & J(z(t_{\mathrm{f}})) \\
\text{s.a} \quad & \dot{z} = f(z,u,\theta),\ z(0) = z_0 \\
& h(z,u,\theta) = 0 \\
& g(z,u,\theta) \leq 0 \\
& h_{\mathrm{f}}(z) = 0 \\
& g_{\mathrm{f}}(z) \leq 0
\end{aligned}
$$

$$
\mathscr{L} = J(z_{\mathrm{f}},t_{\mathrm{f}}) + \mu_{\mathrm{f}}^{\mathrm{T}} g_{\mathrm{f}} + \nu_{\mathrm{f}}^{\mathrm{T}} h_{\mathrm{f}} + \int_0^{t_{\mathrm{f}}} \left(\lambda(t)^{\mathrm{T}} \left(f(z,u,\theta) - \dot{z} \right) + \mu^{\mathrm{T}}(t) g(z,u,\theta) + \nu^{\mathrm{T}}(t) h(z,u,\theta) \right) dt
$$

Necessary conditions:

$$\frac{\partial \mathscr{L}}{\partial z(t)}: \qquad \frac{\partial f}{\partial z}\lambda + \dot{\lambda}(t) + \frac{\partial g}{\partial z}\mu + \frac{\partial h}{\partial z}\nu = 0 \quad \text{(adjoint equations)}$$

$$\frac{\partial \mathscr{L}}{\partial u(t)}: \qquad \frac{\partial f}{\partial u}\lambda + \frac{\partial g}{\partial u}\mu + \frac{\partial h}{\partial u}\nu = 0$$

$$\frac{\partial \mathscr{L}}{\partial z_{\mathrm{f}}}: \qquad \frac{\partial J}{\partial z_{\mathrm{f}}} + \frac{\partial g_{\mathrm{f}}}{\partial z_{\mathrm{f}}}\mu_{\mathrm{f}} + \frac{\partial h_{\mathrm{f}}}{\partial z_{\mathrm{f}}}\nu_{\mathrm{f}} - \lambda(t_{\mathrm{f}}) = 0 \quad \text{(transversality)}$$

Dificulties:

- **nonlinear TPBVP** ($z(t)$, $\lambda(t) \in \mathbb{R}^n$).
- Initialization, **path constraints**.

General algorithms (e.g. CVI) difficult to implement. However analytical solutions for **special cases**. (e.g., linear — Kalman).

5.2.2. Sequential solution and optimization

DAE model treated as an indivisible block (***black box***), solved by a specific routine.

- **IVP**.
- **BVP**, can require the use of **shooting** or **multiple shooting**.

Other characteristics:

- Optimization iterates are **always feasible** solutions of the DAE system (advantageous for some applications).
- Size of NLP generated are also **smaller** than simultaneous approach.
- Sequential method is only reliable when model equations contain only **stable modes**.
- **Bounds on state variables difficult to specify** (hidden). However, can penalize several measures of constraint violation in the objective.

Gradient information can be computed by:

- **Perturbation**:

$$\frac{\partial \phi(t_F)}{\partial \theta} = \frac{\phi(t_F, \theta + \Delta\theta) - \phi(t_F, \theta)}{\Delta\theta}$$

  - ▷ Simple to use.
  - ▷ Choose $\Delta\theta$, considering accuracy of integration.
  - ▷ $n_p + 1$ integrations required.

- **Sensitivities**: Define

$$s(t) = \frac{\partial z(t)}{\partial \theta}$$

Differentiate model equations:

$$\begin{cases} \dfrac{\partial}{\partial \theta}\left[\dot{z} = f(z,\theta)\right] \\ \dfrac{\partial}{\partial \theta}\left[z(0) = z_0\right] \end{cases} \quad \Rightarrow \quad \begin{cases} \dot{s} = \dfrac{\partial f^{\mathrm{T}}}{\partial z} s + \dfrac{\partial f}{\partial \theta} \\ s(0) = \dfrac{\partial z_0}{\partial \theta} \end{cases}$$

  - ▷ **Linear system**, also IVP.
  - ▷ Jacobian already available, in implicit methods.

Then

$$\frac{\partial \phi(z(t_F))}{\partial \theta} = \left.\frac{\partial \phi(z)}{\partial z}\right|_{z_F} \cdot s(t_F)$$

- **Adjoint equations** (Sargent and Sulivan, 1979): Similarly to previous analysis, define

$$\phi(t_F) = \int_0^{t_F} \lambda^{\mathrm{T}}(\dot{z} - f(z,\theta))\,dt$$

Integrate by parts, and set variations to 0:

$$\dot{\lambda} = -\frac{\partial f}{\partial z}\lambda \qquad \text{adjoint equations}$$

$$\lambda(t_F) = \frac{\partial \phi}{\partial z(t_F)} \qquad \text{final condition}$$

$$\frac{\partial \phi(t_F)}{\partial \theta} = \int_0^{t_F} \lambda^{\mathrm{T}} \frac{\partial f}{\partial \theta} dt + \lambda(0)\frac{\partial z_0}{\partial \theta}$$

  - ▷ Now **BVP**, solution can be decoupled, storing trajectory and Jacobian.

5.2.3. Simultaneous solution and optimization

DAE model converted to AE model, by parametrization of the input and state profiles:

- Solve model just **once**, to optimality. If algorithms fails, ... (**infeasible path** approach).
- Originates **large-scale NLPs**.
- Can handle more easily **path constraints**.
- Based on a method for **solution of the DAE system**:

Relaxation methods (finite differences):

$$\dot{z} = f(t,z)$$
$$g(z(a),z(b)) = 0$$

Classical **2nd order** schemes:

- **Trapezoidal**:

$$\frac{z_{i+1}-z_i}{h_i} = \frac{1}{2}\left(f(t_{i+1},z_{i+1})+f(t_i,z_i)\right), \qquad i=1,\ldots,N$$
$$g(z_1,z_{N+1})=0$$

- **Midpoint**:

$$\frac{z_{i+1}-z_i}{h_i} = f\left(t_{i+\frac{1}{2}},\frac{1}{2}(z_i+z_{i+1})\right), \qquad i=1,\ldots,N$$

- **Sparse** system of NL equations.
- Higher order schemes can be used, e.g., **implicit Runge-Kutta** (Ascher et al., 1995).
- **Adaptive grids** can also be used.

Method of weighted residuals:

For $\dot{z} = f(t,u,z,\theta), \qquad z(0) = z_0$

approximate solution:

$$z(t) \simeq z_N(t) = z_0(t) + \sum_i a_i \phi_i(t)$$

Residuals of approximation:

$$R(a,t) = z_N(t) - f(t,u,z_N,\theta)$$

Solve **integral error** equations:

$$E(w_i,R) = \int_0^{t_F} w_i(t) R(a,t)\, dt = 0, \qquad i = 1,\ldots,N$$

For particular choices of $w_i(t)$, methods of **least squares**, **Galerkin**, or **collocation** result.

A common choice is **orthogonal collocation** at **roots of Legendre polynomial**, using **Lagrange interpolation** formulas:

- **Lagrange interpolation polynomials:**

$$z(t) = \sum_{k=0}^{n_k} z_k l_k(t) \qquad u(t) = \sum_{k=0}^{n_k} u_k l_k(t)$$

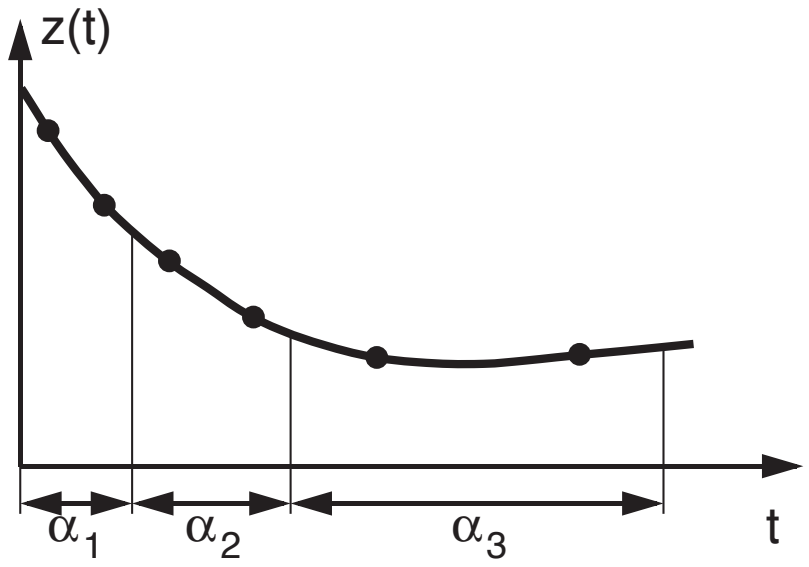
$$l_j(t) = \prod_{\substack{k=0 \\ k \neq j}}^{n_k} \frac{(t - t_k)}{(t_j - t_k)}$$

- **Orthogonal collocation** at t_i:

$$R(a, t_i) = \sum_i z_i l_j'(t_i) - f(z_i, u_i, \theta) = 0$$

- t_i chosen as **roots of Legendre polynomial** of degree n, shifted to $t \in [0, t_F]$.
  This can be shown to be equivalent to a Runge-Kutta method of $O(2n_c)$.

- Can also be extended to collocation of **finite elements**:



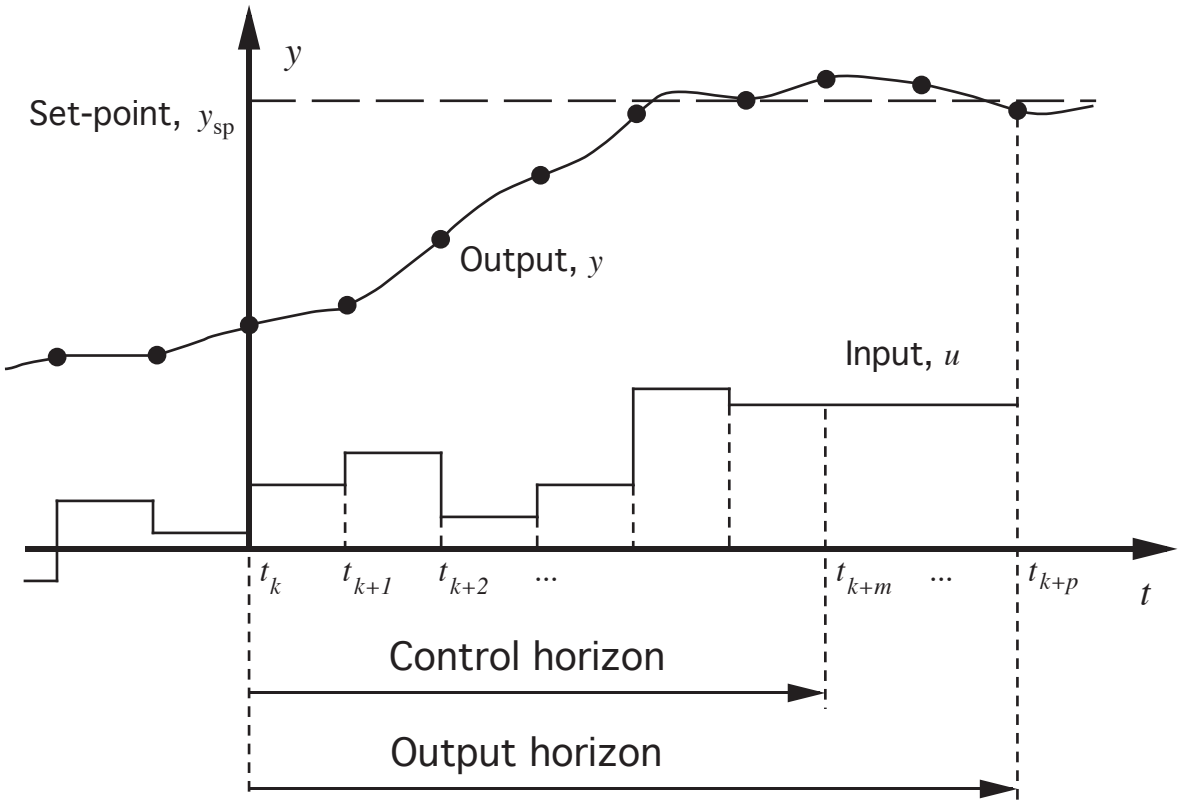
  - ▷ Better control of **approximation error**:

$$\|e(\alpha_{i+1})\| = O(|\Delta\alpha_i\|^{2n_c})$$

    Can be used to control the element length, to **equidistribute** the solution error.
  - ▷ Include **continuity** of solution and derivatives across elements.

5.3. Process applications

5.3.1. Nonlinear model-predictive control



- Requires a **process model**.
- Based on **online optimization**.
- **feedback / feedforward** structure.
- can also be adapted to **observation**, **parameter estimation**.

Examples:

- **QDMC** — linear convolution model, quadratic objective, QP (Garcia and Morshedi, 1986).
- **Newton-control** — nonlinear model, quadratic objective, NLP (Oliveira and Biegler, 1995).

Hierarchical process supervision

Disturbance type:

Long term:

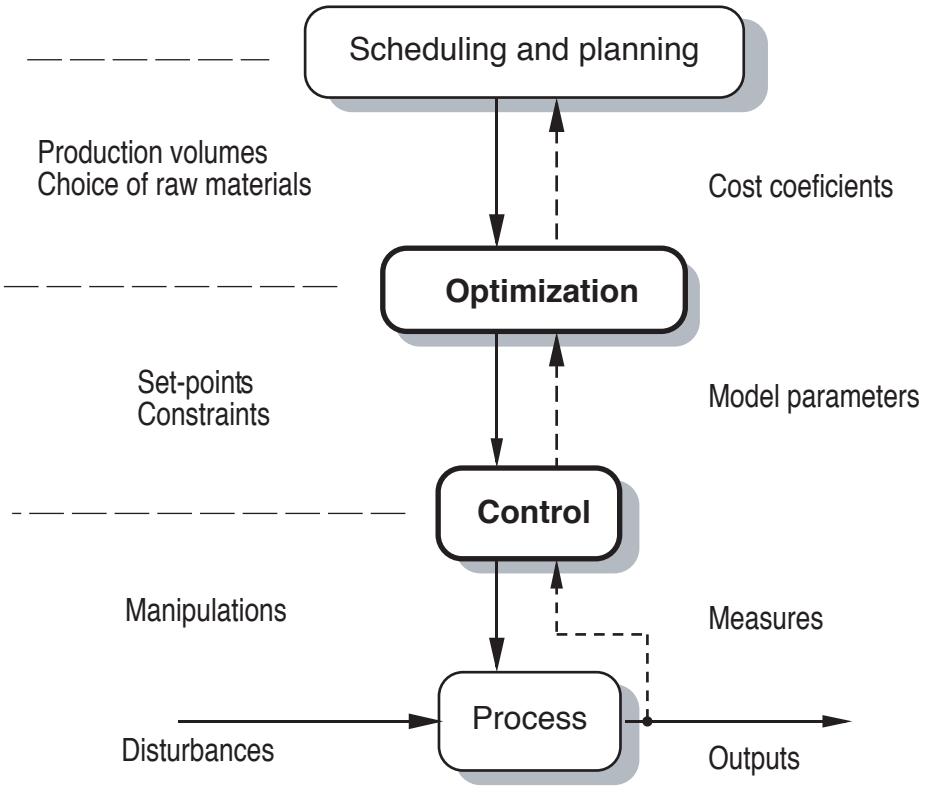
Market fluctuations

Mean term:

Price fluctuations
Catalyst deactivation
Raw materials

Short term:

Process variations



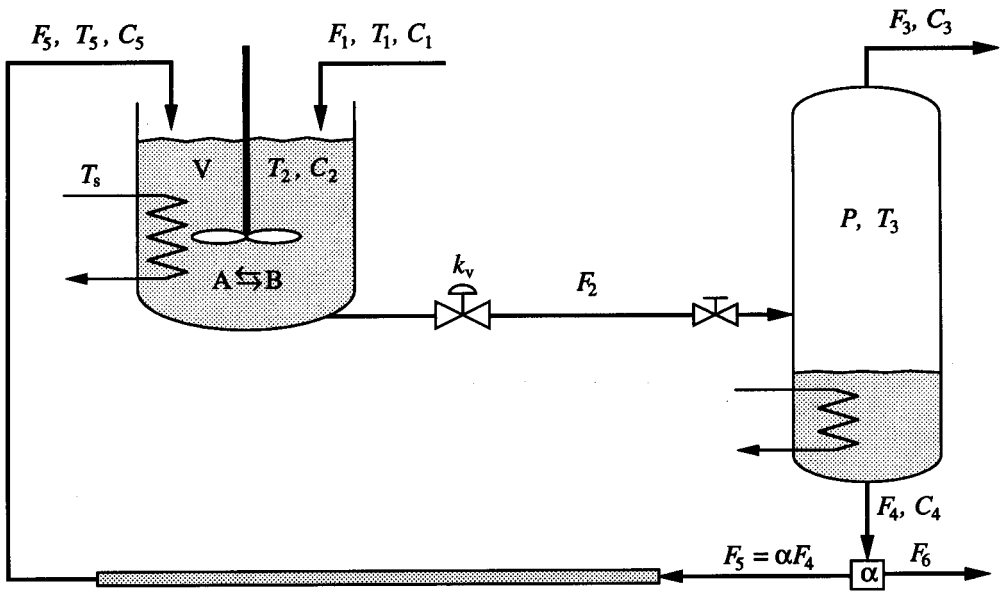
Problem formulation

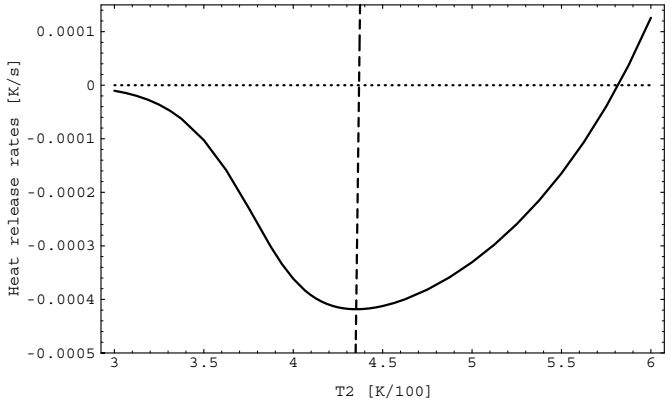
$$\min_{u(t)\in\mathscr{H}_{ik}} \quad J_2(u,y) = \int_{t_k}^{t_k+t_{oh}} (y-y_{sp})^{\mathrm{T}} Q_y(t)(y-y_{sp}) + (u-u_r)^{\mathrm{T}} Q_u(t)(u-u_r)\,dt$$

$$\begin{aligned} \text{s.t.} \quad \dot{x} &= f_p(x,u,d;\theta) \\ y &= g_p(x;\theta) \\ u_l &\le u \le u_u \\ x_l &\le x \le x_u \\ y_l &\le y \le y_u \end{aligned}$$

- u_r is **input reference** trajectory.
- If $t_{ih} \le t_{oh}$, assume inputs remain constant.

Flash and CSTR example:





Reactor equations:

$$\frac{dV}{dt} = F_1 + F_5 - F_2$$
$$\frac{dC_2}{dt} = \frac{F_1 C_1 + F_5 C_4 - (F_1 + F_5) C_2}{V} + r(T_2, C_2)$$
$$\frac{dT_2}{dt} = \frac{F_1 T_1 + F_5 T_3 - (F_1 + F_5) T_2}{V} + \frac{(-\Delta H)}{\rho c_p} r(T_2, C_2) - \frac{UA}{\rho c_p V}(T_2 - T_s)$$

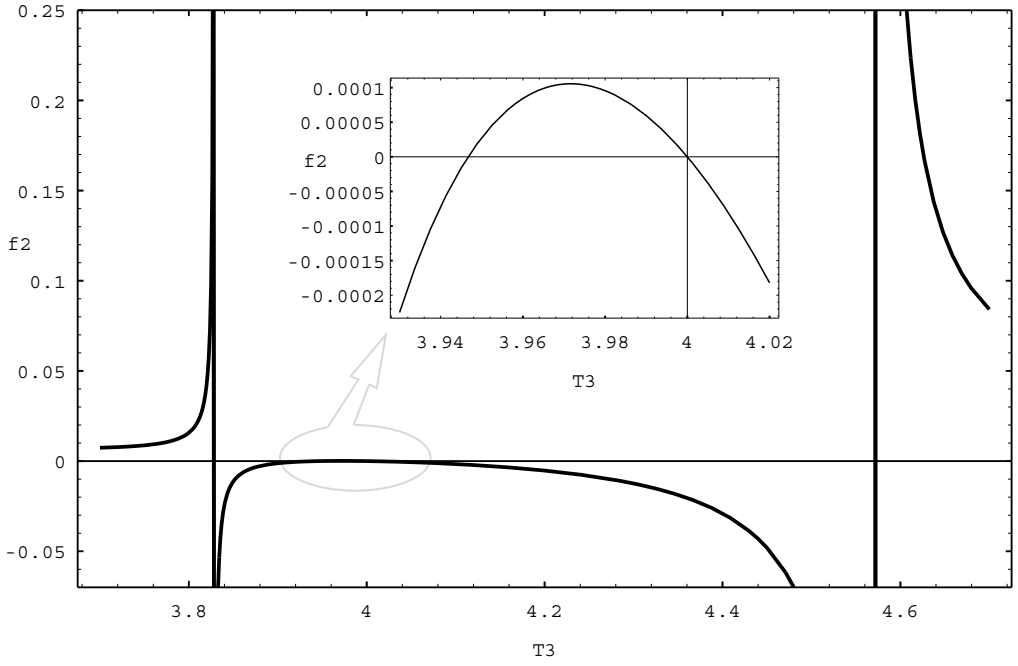
Flash equations:

$$C_3 = K_{e1}(T_3) C_4$$
$$(1 - C_3) = K_{e2}(T_3)(1 - C_4)$$
$$F_2 = F_3 + F_4$$
$$F_2 C_2 = F_3 C_3 + F_4 C_4$$

Mixed equations:

$$r(T_2, C_2) = -K_{r1}(T_2) C_2 + K_{r2}(T_2)(1 - C_2)$$
$$K_{r1} = A_1 \exp(-E_{a1}/T_2)$$
$$K_{r2} = A_2 \exp(-E_{a2}/T_2)$$
$$K_{e1}(T_3) = a_1 10^{-b_1/T_3}$$
$$K_{e2}(T_3) = a_2 10^{-b_2/T_3}$$
$$F_2 = k_v V$$
$$C_3 + C_4 = 1$$
$$F_5 = \alpha F_4$$

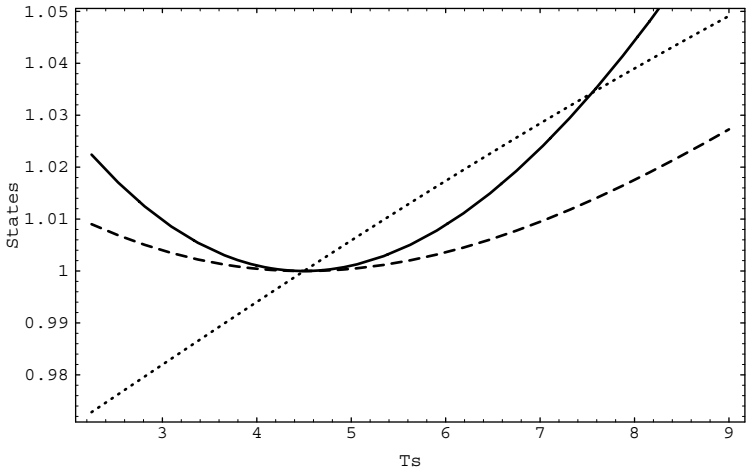
Input multiplicities:

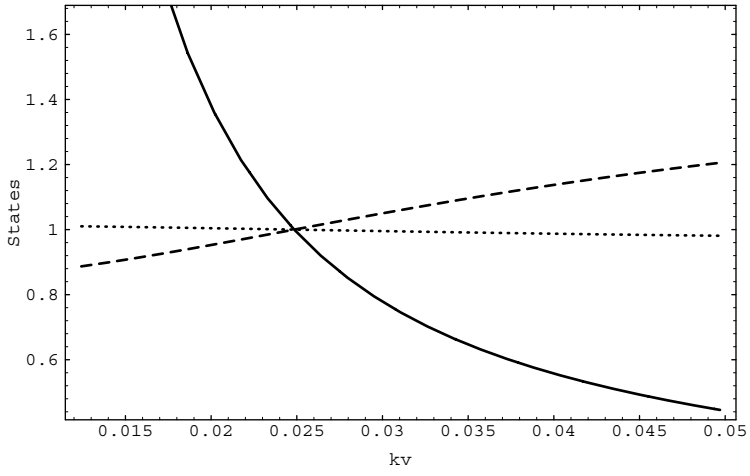


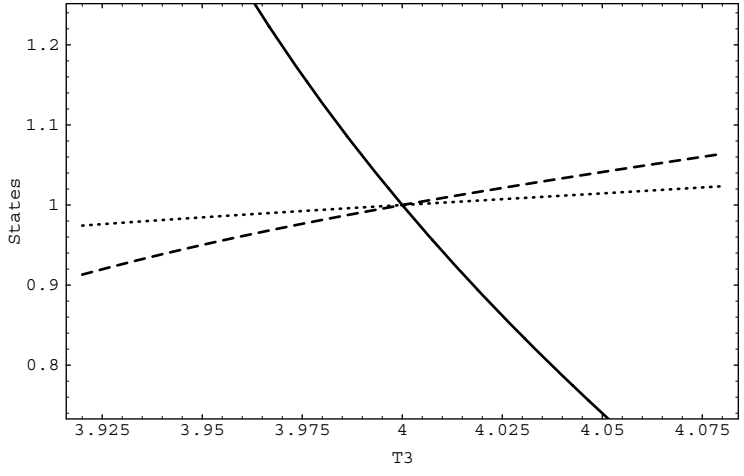
$$u = \begin{cases} \begin{bmatrix} 4.500 & 2.485 \times 10^{-2} & 4.000 \end{bmatrix}^{\mathrm{T}} \\ \begin{bmatrix} 6.861 & 3.179 \times 10^{-2} & 3.947 \end{bmatrix}^{\mathrm{T}} \end{cases} \qquad \Rightarrow \qquad x = \begin{bmatrix} 1 & 0.4899 & 4.352 \end{bmatrix}^{\mathrm{T}}$$

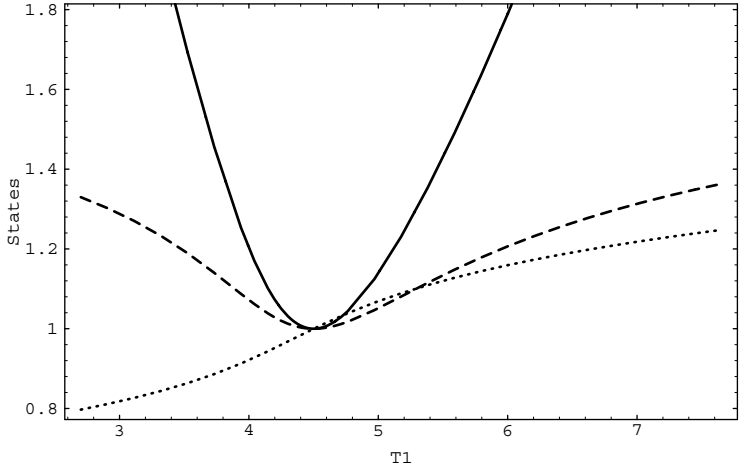
- $\neq$ manipulations have significant economic impact.
- linear control can originate **sudden instability**.

Steady state sensitivities:

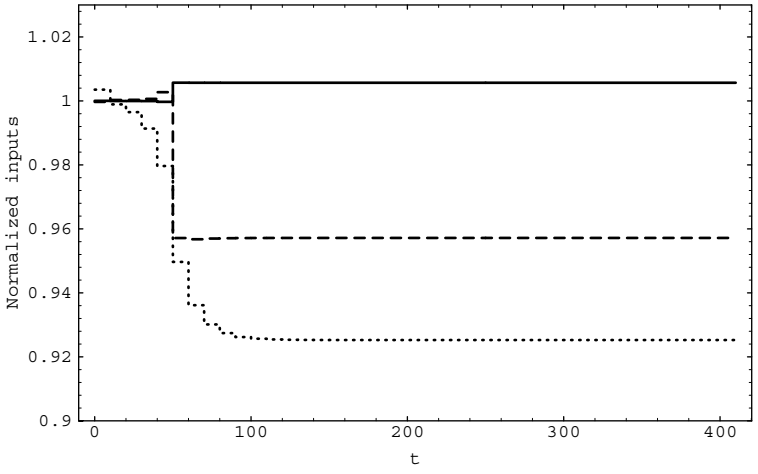


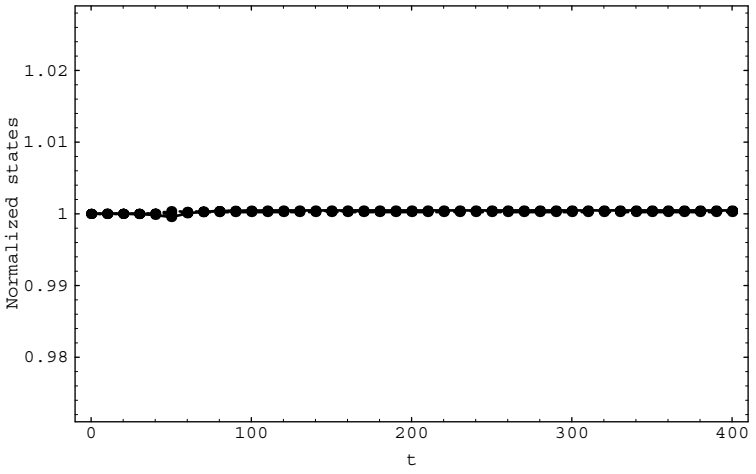






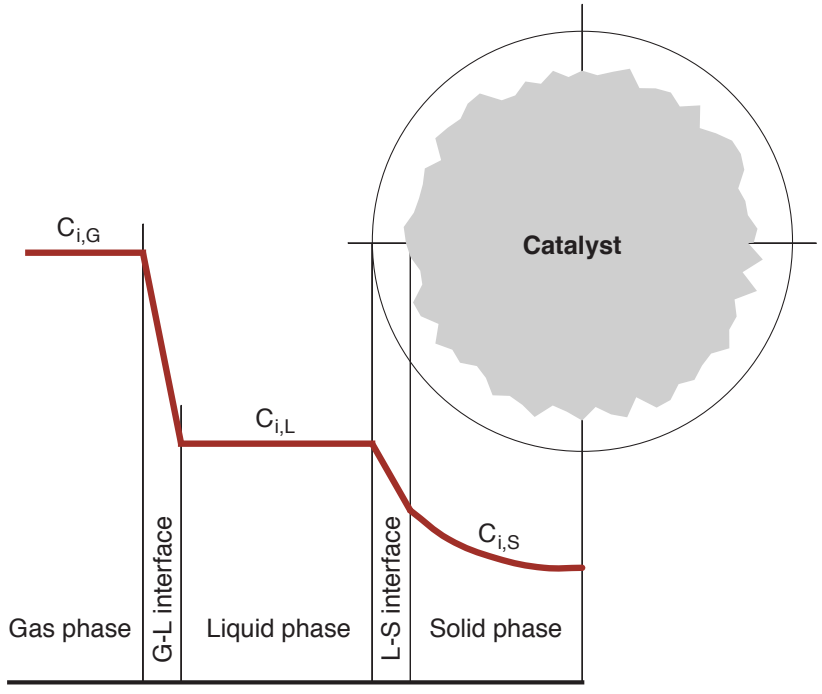
Schedule change in recycle ratio:





5.3.2. Optimal design of catalyst pellets

Catalytic reactions very common in ChemE:



Active materials generally expensive. Where to concentrate activity?

Non-uniform distribution can lead to improved performance (e.g., **efficiency**, **selectivity** and **duration**).

- Significant no. of cases studied. Sometimes solution is (e.g., Chemburkar et al. (1987)):

$$a(r) = \delta(r - r^*)$$

- Need to consider detailed information, e.g., catalyst **poisoning** or **coking**.

Here consider **optimal activity profiles**, for pellets with **single arbitrary reaction**, without catalyst deactivation:

Objective: Maximize **effectiveness factor**, using **simultaneous solution and optimization**.

Mathematical model:

$$\max_{a(r)} \quad (n+1)\int_0^1 \psi(c,\theta)a(x)x^n\,dx$$

$$\text{s.t.} \quad \frac{1}{x^n}\frac{d}{dx}\left(x^n\frac{dc}{dx}\right) = (n+1)\phi^2 a(x)\psi(c,\theta)$$

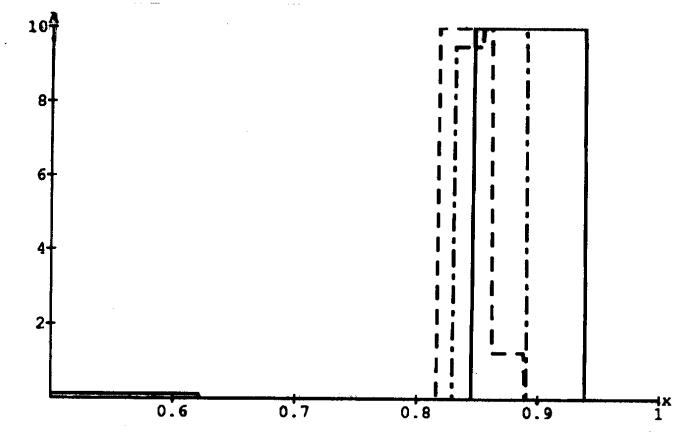
$$\frac{1}{x^n}\frac{d}{dx}\left(x^n\frac{d\theta}{dx}\right) = -(n+1)\beta\phi^2 a(x)\psi(c,\theta)$$

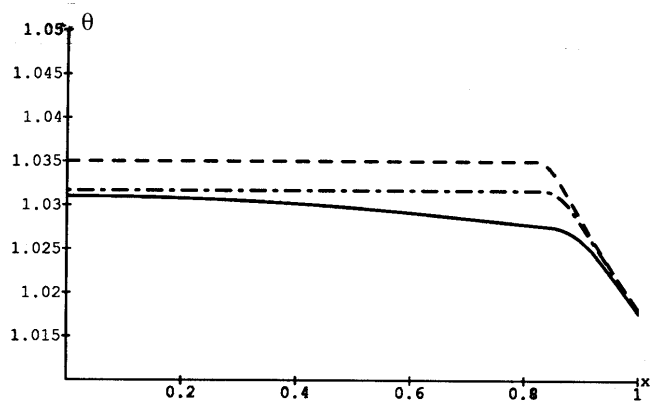
$$\int_0^1 a(x)x^n\,dx = \frac{1}{(n+1)}$$

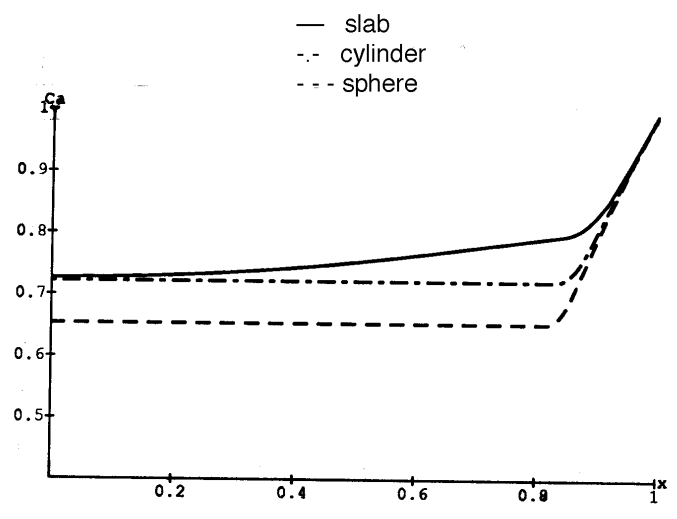
$$x=0 \quad \Rightarrow \quad \frac{dc}{dx} = \frac{d\theta}{dx} = 0$$

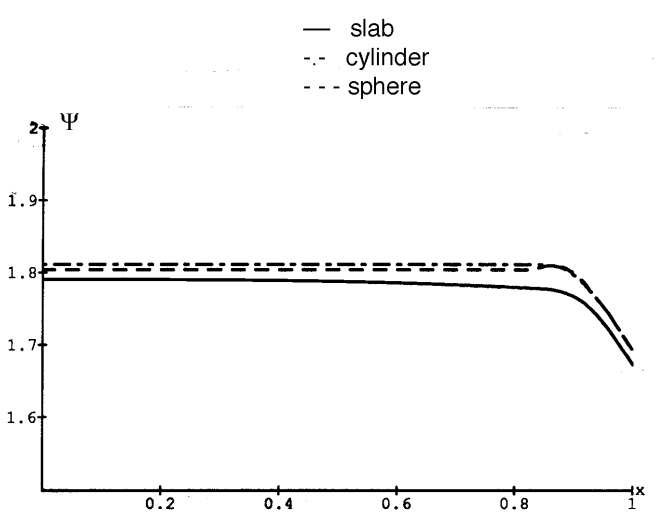
$$x=1 \quad \Rightarrow \quad \frac{dc}{dx} = \mathrm{Bi_m}(1-c); \quad \frac{d\theta}{dx} = \mathrm{Bi_t}(1-\theta)$$

Optimization results:









- **Orthogonal collocation** with $n_k = 3$, using 20 finite elements.
- Physical data:

β	γ	ϕ	Bi_m	Bi_{th}	a_{max}
0.05	30	1	250	5	10

- Model results:

Geometry (file)	Active catalyst location	η_1 (constant a)	η_{opt}	Increase (%)	η_δ^* (estimated)
Slab(6)	0.85–0.94	1.472	1.768	20.1	1.672
Cylinder(1)	0.83–0.89	1.578	1.809	14.6	1.692
Sphere(2)	0.82–0.86	1.639	1.807	10.3	1.691

- **Multiple solutions** possible!

- Empirical strategy for **element length adaptation**:

$$\text{If} \quad \left| \frac{\max_i \text{Rnc}_i}{\min_i \text{Rnc}_i} \right| \geq \xi \quad \text{then} \quad \Delta\alpha_{i,\text{new}} = \frac{\Delta\alpha_{i,\text{old}}}{\|\text{Rnc}_i\|^{\varepsilon}}$$

  - ▷ $\eta \sim 0.2$, 5 or 6 NLP iterations required.
  - ▷ Only 2–3 elements active at solution.

- **Example code** available at
  <http://www.eq.uc.pt/~nuno/cim2003/fopslab6.gms>.

Conclusions

- **Representations** and **models** are the natural tools in PSE.
- **Systematic methods** are required to explore alternatives that meet constraints and optimize objective. Their use depend significantly on **key** model characteristics.
- PSE has both **practical** and **theoretical** implications.
- Current research areas also at **micro-scale** (molecular simulation, CFD, product design) and **macro-scale** (logistics, production planning), and **integration** of all levels.

References

U.M. Ascher, R.M.M. Mattheij, R.D. Russel (1995). *Numerical Soltution of Boundary Value Problems for Ordinary Differential Equations*, SIAM, Philadelphia, PA.

B.M. Bhatti (2000). *Practical Optimization Methods with Mathematica Applications*, Springer-Verlag, Berlin.

L.T. Biegler, I.E. Grossmann, A.W. Westerberg (1997). *Systematic Methods of Chemical Process Design*, Prentice-Hall PTR, Upper Saddle River, NJ.

L.T. Biegler, A.M. Cervantes, A. Wächter (2002). "Advances in Simultaneous Strategies for Dynamic Process Optimization", *Chem. Eng. Sci.*, **57**(4), 575–593.

R.B. Bird, W.E. Stewart, E.N. Lightfoot, *Transport Phenomena*, 2nd Edition, J. Wiley & Sons, New York.

A. Brooke, D. Kendrick, A. Meeraus (1992). *Gams: A User's Guide*, The Scientific Press, San Francisco.

M.R. Bussieck, A.S. Drud (2001). "SBB: A New Solver for Mixed Integer Nonlinear Programming", <http://www.gams.com/presentations/or01/sbb.pdf>.

R.M. Chemburkar, M. Morbidelli, A. Varma, "Optimal Catalyst Activity in Pellets - VII", *Chem. Eng. Sci.*, **42**(11), 2621–2632.

J.E. Cuthrell, L.T. Biegler (1987). "On the Optimization of Differential-algebraic Process Systems", *AIChE J.*, **33**, 1257–1270.

J.E. Dennis, R.B. Schnabel (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall Int., Englewood Cliffs, NJ.

J.M. Douglas (1988). *Conceptual Design of Chemical Processes*, McGraw-Hill, New York.

C.A. Floudas (1995). *Nonlinear and Mixed-Integer Optimization, Theory and Applications*, Oxford University Press, New York.

R. Fourer, D.M. Gay, B.W. Kernighan (1993). *AMPL, A Modeling Language for Mathematical Programming*, Scientific Press.

F. Friedler, K. Tarjan, Y.W. Huang, L.T. Fan (1993). "Graph-Theoretic Approach to Process Synthesis: Polynomial Algorithm for Maximal Structure Generation", *Comp. & Chem. Eng.*, **17**, 929–942.

F. Friedler, J.B. Varga, L.T. Fan (1995). "Decision-Mapping: A Tool for Consistent and Complete Decisions in Process Synthesis", *Chem. Eng. Sci.*, **50**, 1755–1768.

C.E. Garcia, A.M. Morshedi (1986). "Quadratic Programming Solution of Dynamic Matrix Control (QDMC)", *Chem. Eng. Commun.*, **46**, 73–87.

P.E. Gill, W. Murray, M.H. Wright, *Practical Optimization*, Academic Press, London.

L. Grippo, F. Lampariello, S. Lucidi (1986). "A Nonmonotone Line Search Technique for Newton's Method", *SIAM J. Num. Anal.* **23**, 707–716.

I.E. Grossmann, A.W. Westerberg (2000). "Research Challenges in Process Systems Engineering", *AIChE J.*, **46**(9), 1700–1703.

I.E. Grossmann (2001). "Review of Non-Linear Mixed Integer and Disjunctive Programming Techiques for Process Systems Engineering", <http://egon.cheme.cmu.edu/Group/Papers/MINLPOPTE.pdf>.

I.E. Grossmann (2004). "Challenges in the New Millennium: Product Discovery and Design, Enterprise and Supply Chain Optimization, Global Life Cycle Assessment", *Proc. PSE2003*.

B. Linnhoff, D.W. Townsend, D. Boland, G.F. Hewitt, B.E.A. Thomas, A.R. Guy, R.H. Marsland (1982). *A User Guide on Process Integration for the Efficient Use of Energy*, The Institution of Chemical Engineers, Rugby, UK.

W. Marquardt, L.v. Wedel, B. Bayes (2000). "Perspectives on Lifecycle Process Modeling", in M.F. Malone, J.A. Trainham, B. Barnahan (Eds.), *AIChE Symp. Ser. 323*, **96**, 192–214.

Z. Michalewicz, D.B. Fogel (2002). *How to Solve It: Modern Heuristics*, Springer-Verlag, Berlin.

J. Nocedal, S.J. Wright (1999). *Numerical Optimization*, Springer-Verlag, Berlin.

N.M.C. Oliveira, L.T. Biegler (1995). "An Extension of Newton-type Algorithms for Nonlinear Process Control", *Automatica*, **31**(2), 281–286.

M.S. Peters, K.D. Timmerhaus, R.E. West (2002). *Plant Design and Economics for Chemical Engineers*, 5th edition, McGraw-Hill, New York.

R. Raman, I.E. Grossmann (1991). "Relation Between MILP Modeling and Logical Inference for Chemical Process Synthesis", *Comp. & Chem. Eng.*, **15**(2), 73–84.

D.F. Rudd, C.C. Watson (1968). *Strategy of Process Engineering*, J. Wiley & Sons, New York.

H.S. Ryoo, N.V. Sahinidis (1995). “Global Optimization of Nonconvex NLPs and MINLPs with Applications in Process Design”, *Comp. & Chem. Eng.*, **19**, 551–566.

N.V. Sahinidis, I.E. Grossmann (1991). “Design of a Chemical Complex”, in I.E. Grossmann (Ed.), *CACHE Design Case Study Volume 6: Chemical Engineering Optimization Models with GAMS*, CACHE Corp., Austin, TX.

R.W.H. Sargent, G.R. Sullivan (1979). “Development of Optimal Feed Changeover Policies for Refinery Distillation Units”, *Ind. & Eng. Chem. Proc. Des. Dev.*, **18**(1), 113.

R.W.H. Sargent (1991). “What is Chemical Engineering?”, *CAST Newsletter*, **14**(1), 9–11.

W.D. Seider, J.D. Seader, D.R. Lewin (1999). *Process Design Principles. Synthesis, Analysis and Evaluation*, J. Wiley & Sons, New York.

U.V. Shenoy (1995). *Heat Exchanger Network Synthesis. Process Optimization by Energy and Resource Analysis*, Gulf Publishing Company, Houston, TX.

R. Turton, R.C. Bailie, W.B. Whiting, J.A. Shaeiwitz (1998). *Analysis, Synthesis, and Design of Chemical Processes*, Prentice-Hall PTR, Upper Saddle River, NJ.

A.W. Westerberg, H.P. Hutchison, R.L. Motard, P. Winter (1979). *Process Flowsheeting*, Cambridge University Press, Cambridge.

A.W. Westerberg (1998). “Process Engineering II — A System’s View”, AIChE Meeting, Miami, FL.

H.P. Williams (1993). *Model Building in Mathematical Programming*, 3rd Edition, J. Wiley & Sons, New York.