Optimization of Differential-Algebraic Equation Systems

L. T. Biegler
Chemical Engineering Department
Carnegie Mellon University
Pittsburgh, PA

DAE Optimization Outline

I Introduction
- Process Examples

II Parametric Optimization
- Gradient Methods
  - Perturbation
  - Direct - Sensitivity Equations
  - Adjoint Equations

III Optimal Control Problems
- Optimality Conditions
- Model Algorithms
  - Sequential Methods
  - Multiple Shooting
  - Indirect Methods

IV Simultaneous Solution Strategies
- Formulation and Properties
- Process Case Studies
- Software Demonstration

2

Dynamic Optimization Problem

$$\min \quad \Phi\left(z(t),\ y(t),\ u(t),\ p,\ t_f\right)$$

$$s.t. \quad \frac{dz(t)}{dt} = f\left(z(t), y(t), u(t), t, p\right)$$

$$g\left(z(t), y(t), u(t), t, p\right) = 0$$

$$z^o = z(0)$$

$$z^l \le z(t) \le z^u$$

$$y^l \le y(t) \le y^u$$

$$u^l \le u(t) \le u^u$$

$$p^l \le p \le p^u$$

t, time
z, differential variables
y, algebraic variables

t_f, final time
u, control variables
p, time independent parameters

3

DAE Models in Process Engineering

Differential Equations

•Conservation Laws (Mass, Energy, Momentum)

Algebraic Equations

•Constitutive Equations, Equilibrium (physical properties, hydraulics, rate laws)
•Semi-explicit form
•Assume to be index one (i.e., algebraic variables can be solved uniquely by algebraic equations)
•If not, DAE can be reformulated to index one (see Ascher and Petzold)

Characteristics

•Large-scale models – not easily scaled
•Sparse but no regular structure
•Direct linear solvers widely used
•Coarse-grained decomposition of linear algebra

4

Parameter Estimation

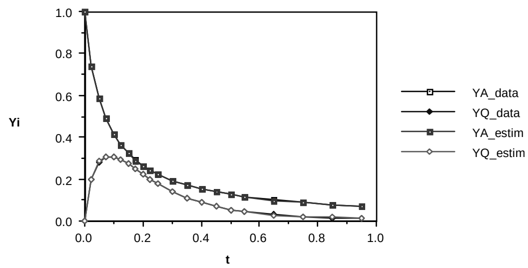
Catalytic Cracking of Gasoil (Tjoa, 1991)

$$A \xrightarrow{p_1} Q,\ Q \xrightarrow{p_2} S,\ A \xrightarrow{p_3} S$$

$$\dot{a} = -(p_1 + p_3)a^2$$

$$\dot{q} = -p_1 a^2 - p_2 q$$

$$a(0) = 1, \quad q(0) = 0$$



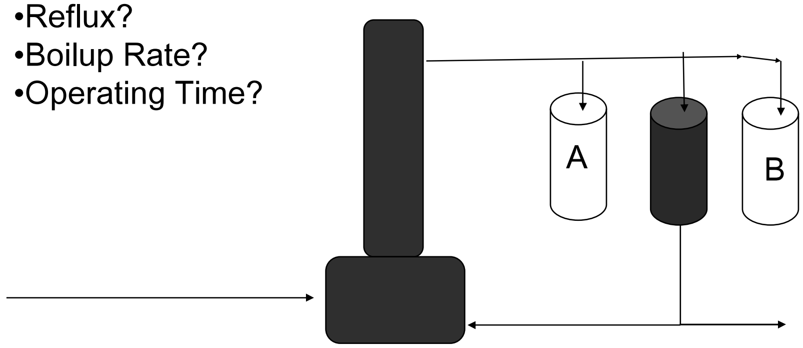
number of states and ODEs: 2
number of parameters:3
no control profiles
constraints: $p_L \leq p \leq p_U$

Objective Function: Ordinary Least Squares

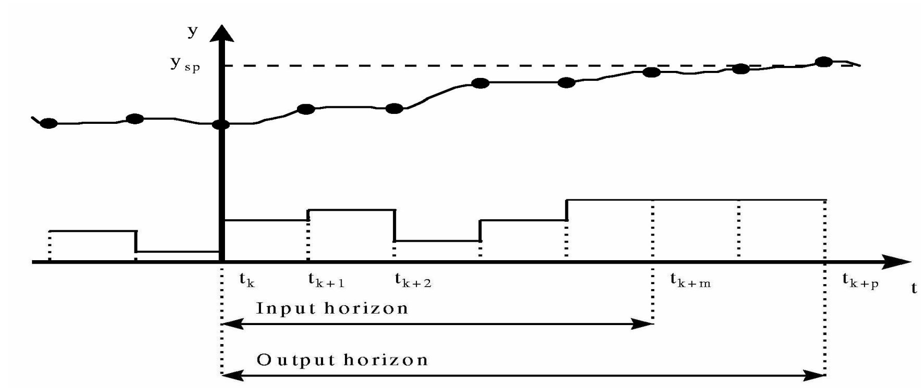
$(p_1, p_2, p_3)^0 = (6, 4, 1)$
$(p_1, p_2, p_3)^* = (11.95, 7.99, 2.02)$
$(p_1, p_2, p_3)_{true} = (12, 8, 2)$

Batch Distillation Multi-product Operating Policies

- Run between distillation batches
- Treat as boundary value optimization problem
  - When to switch from A to offcut to B?
  - How much offcut to recycle?
  - Reflux?
  - Boilup Rate?
  - Operating Time?



Nonlinear Model Predictive Control (NMPC)



$$\min_{\boldsymbol{u}} \quad \sum \| y(t) - y^{sp} \|_{Q^y} + \sum \| \boldsymbol{u}(\boldsymbol{t}^k) - \boldsymbol{u}(\boldsymbol{t}^{k-1}) \|_{Q^u}$$

$$\begin{aligned} s.t. \quad & z'(t) = F(z(t), y(t), \boldsymbol{u}(t), t) \\ & 0 = G(z(t), y(t), \boldsymbol{u}(t), t) \\ & z(t) = z^{\text{init}} \\ & \text{Bound Constraints} \\ & \text{Other Constraints} \end{aligned}$$

7

Batch Process Optimization

Optimization of dynamic batch process operation resulting from reactor and distillation column

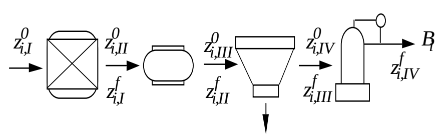
DAE models:

$z' = f(z, y, u, p)$

$g(z, y, u, p) = 0$

$A+B \rightarrow C$
$C+B \rightarrow P+E$
$P+C \rightarrow G$

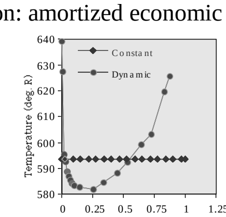
number of states and DAEs: $n_z + n_y$
parameters for equipment design (reactor, column)
n_u control profiles for optimal operation



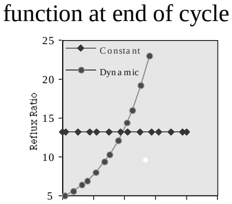
Constraints: $u_L \le u(t) \le u_U$ $\quad z_L \le z(t) \le z_U$

$y_L \le y(t) \le y_U$ $\quad p_L \le p \le p_U$

Objective Function: amortized economic function at end of cycle time t_f



optimal reactor temperature policy



optimal column reflux ratio

8

Reactor Design Example

Plug Flow Reactor Optimization

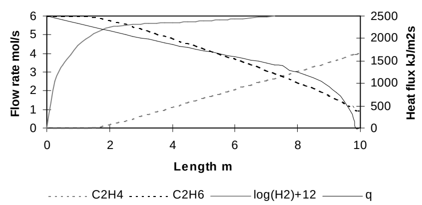
The cracking furnace is an important example in the olefin production industry, where various hydrocarbon feedstocks react. Consider a simplified model for ethane cracking (Chen et al., 1996). The objective is to find an optimal profile for the heat flux along the reactor in order to maximize the production of ethylene.

$$\begin{aligned} \text{Max}\ & F_{exit}^{C_2H_4} \\ \text{s.t.}\ & \text{DAE} \\ T_{exit}\ & \le 1180\text{K} \end{aligned}$$

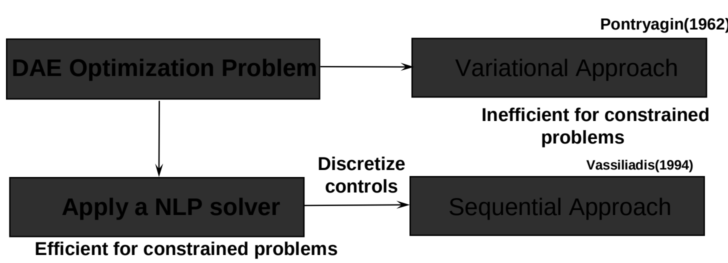
The reaction system includes six molecules, three free radicals, and seven reactions. The model also includes the heat balance and the pressure drop equation. This gives a total of eleven differential equations.

$$C_2H_6 \rightarrow 2CH_3\bullet \qquad CH_3\bullet + C_2H_6 \rightarrow CH_4 + C_2H_5\bullet$$
$$H\bullet + C_2H_6 \rightarrow H_2 + C_2H_5\bullet \qquad C_2H_5\bullet \rightarrow C_2H_4 + H\bullet$$
$$C_2H_5\bullet + C_2H_4 \rightarrow C_3H_6 + CH_3\bullet \qquad 2C_2H_5\bullet \rightarrow C_4H_{10}$$
$$H\bullet + C_2H_4 \rightarrow C_2H_5\bullet$$

Concentration and Heat Addition Profile



Dynamic Optimization Approaches



Sequential Approaches - Parameter Optimization

Consider a simpler problem without control profiles:

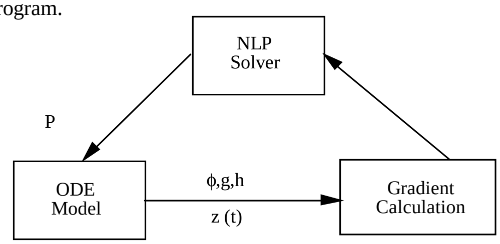
e.g., equipment design with DAE models - reactors, absorbers, heat exchangers

$$\mathit{Min} \quad \Phi(z(t_f))$$

$$z' = f(z, p),\ z(0) = z_0$$

$$g(z(t_f)) \le 0,\ h(z(t_f)) = 0$$

By treating the ODE model as a "black-box" a sequential algorithm can be constructed that can be treated as a nonlinear program.



Task: How are gradients calculated for optimizer?

11

Gradient Calculation

Perturbation

Sensitivity Equations

Adjoint Equations

<u>Perturbation</u>

Calculate approximate gradient by solving ODE model (np + 1) times

Let $\psi = \Phi$, g and h (at $t = t_f$)

$$d\psi/dp_i = \{\psi(p_i + \Delta p_i) - \psi(p_i)\}/\Delta p_i$$

Very simple to set up

Leads to poor performance of optimizer and poor detection of optimum unless roundoff error ($O(1/\Delta p_i)$) and truncation error ($O(\Delta p_i)$) are small.

Work is proportional to np (expensive)

12

Direct Sensitivity

From ODE model: $\frac{\partial}{\partial p}\{z' = f(z, p, t), z(0) = z_0(p)\}$

$$\text{define } s_i(t) = \frac{\partial z(t)}{\partial p_i} \; i = 1, \ldots \text{np}$$

$$s_i' = \frac{d}{dt}(s_i) = \frac{\partial f}{\partial p_i} + \frac{\partial f}{\partial z}^T s_i, \;\; s_i(0) = \frac{\partial z(0)}{\partial p_i}$$

(nz x np sensitivity equations)

- z and s_i , i = 1,…np, an be integrated forward simultaneously.
- for implicit ODE solvers, $s_i(t)$ can be carried forward in time after converging on z
- linear sensitivity equations exploited in ODESSA, DASSAC, DASPK, DSL48s and a number of other DAE solvers

Sensitivity equations are efficient for problems with many more constraints than parameters (1 + ng + nh > np)

13

Example: Sensitivity Equations

$$z_1' = z_1^2 + z_2^2$$
$$z_2' = z_1 z_2 + z_1 p_b$$
$$z_1 = 5, z_2(0) = p_a$$
$$s(t)_{a,j} = \partial z(t)_j / \partial p_a, s(t)_{b,j} = \partial z(t)_j / \partial p_b, \; j = 1,2$$
$$s_{a,1}' = 2z_1 s_{a,1} + 2z_2 s_{a,2}$$
$$s_{a,2}' = z_1 s_{a,2} + z_2 s_{a,1} + s_{a,1} p_b$$
$$s_{a,1} = 0, s_{a,2}(0) = 1$$

$$s_{b,1}' = 2z_1 s_{b,1} + 2z_2 s_{b,2}$$
$$s_{b,2}' = z_1 + z_1 s_{b,2} + z_2 s_{b,1} + s_{b,1} p_b$$
$$s_{b,1} = 0, s_{b,2}(0) = 0$$

14

Adjoint Sensitivity

Adjoint or Dual approach to sensitivity

Adjoin model to objective function <u>or</u> constraint

($\psi = \Phi$,g or h)

$$\psi = \psi(t_f) - \int_0^{t_f} \lambda^T (z' - f(z, p, t))dt$$

(λ(t)) serve as multipliers on ODE's)

Now, integrate by parts

$$\psi = \psi(t_f) + \lambda(0)^T z_0(p) - \lambda(t_f)^T z(t_f) + \int_0^{t_f} (z^T \lambda' + \lambda^T f(z, p, t))dt$$

$$d\psi = \cancelto{0}{\left[\frac{\partial \psi(z(t_f))}{\partial z(t_f)} - \lambda(t_f)\right]} \delta z(t_f) + \left[\frac{\partial z_0(p)}{\partial p}\lambda(0)\right]^T dp + \int_0^{t_f} \cancelto{0}{\left[\lambda' + \frac{\partial f}{\partial z}\lambda\right]^T} \delta z(t) + \left[\frac{\partial f}{\partial p}\lambda\right]^T dp\, dt$$

Take variations and find $d\psi/dp$ subject to feasibility of ODE's

Now, set all terms <u>not</u> in dp to zero.

Adjoint System

$$\lambda' = -\frac{\partial f}{\partial z}\lambda(t), \;\; \lambda(t_f) = \frac{\partial \psi(z(t_f))}{\partial z(t_f)}$$

$$\frac{d\psi}{dp} = \frac{\partial z_0(p)}{\partial p}\lambda(0) + \int_0^{t_f} \left[\frac{\partial f}{\partial p}\lambda(t)\right] dt$$

Integrate model equations <u>forward</u>

Integrate adjoint equations backward and evaluate integral and sensitivities.

Notes:

nz (ng + nh + 1) adjoint equations must be solved backward (one for each objective and constraint function)

for implicit ODE solvers, profiles (and even matrices) can be stored and carried backward after solving forward for z as in DASPK/Adjoint (Li and Petzold) and CVODES (Serban and Hindmarsh)

more efficient on problems where: np > 1 + ng + nh

Example: Adjoint Equations

$$z_1' = z_1^2 + z_2^2$$

$$z_1' = z_1 z_2 + z_1 p_b$$

$$z_1 = 5, z_2(0) = p_a$$

$$\text{Form} \quad \lambda^T f(z,p,t) = \lambda_1(z_1^2 + z_2^2) + \lambda_2(z_1 z_2 + z_1 p_b)$$

$$\lambda' = -\frac{\partial f}{\partial z}\lambda(t), \quad \lambda(t_f) = \frac{\partial \psi(z(t_f))}{\partial z(t_f)}$$

$$\frac{d\psi}{dp} = \frac{\partial z_0(p)}{\partial p}\lambda(0) + \int_0^{t_f}\left[\frac{\partial f}{\partial p}\lambda(t)\right]dt$$

then becomes :

$$\lambda_1' = -2\lambda_1 z_1 - \lambda_2(z_2 + p_b), \quad \lambda_1(t_f) = \frac{\partial \psi(t_f)}{\partial z_1(t_f)}$$

$$\lambda_2' = -2\lambda_1 z_2 - \lambda_2 z_1, \quad \lambda_2(t_f) = \frac{\partial \psi(t_f)}{\partial z_2(t_f)}$$

$$\frac{d\psi(t_f)}{dp_a} = \lambda_2(0)$$

$$\frac{d\psi(t_f)}{dp_b} = \int_0^{t_f}\lambda_2(t) z_1(t)\,dt$$

17

Example: Hot Spot Reactor

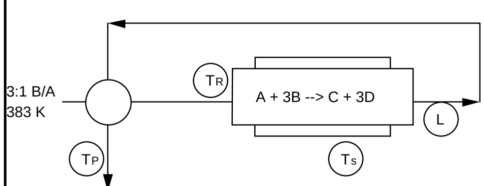
$$\underset{T_P,T_R,L,T_S}{Min} \quad \Phi = L - \int_0^L (T(t) - T_S/T_R)dt$$

$$s.t. \quad \frac{dq}{dt} = 0.3(1-q(t))\exp[20-20/T(t)], \quad q(0)=0$$

$$\frac{dT}{dt} = -1.5(T(t) - T_S/T_R) + 2/3\frac{dq}{dt}, \quad T(0)=1$$

$$\Delta H_{feed}(T_R, 110^o C) - \Delta H_{product}(T_P, T(L)) = 0$$

$$T_P = 120^o C, \quad T(L) = 1 + 10^o C/T_R$$



T_P = specified product temperature
T_R = reactor inlet, reference temperature
L = reactor length
T_S = steam sink temperature
q(t) = reactor conversion profile
T(t) = normalized reactor temperature profile

Cases considered:

- Hot Spot - no state variable constraints
- Hot Spot with $T(t) \le 1.45$

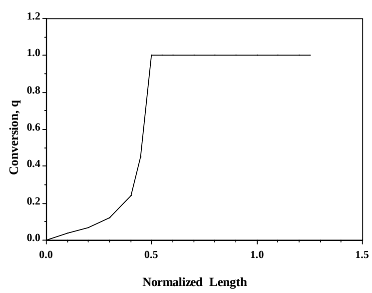
18

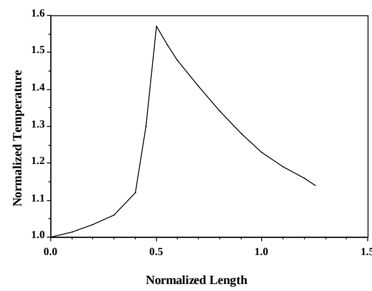
Hot Spot Reactor: Unconstrained Case

Method: SQP (perturbation derivatives)

	L(norm)	T_R(K)	T_S(K)	T_P(K)
Initial:	1.0	462.23	425.26	250
Optimal:	1.25	500	470.1	188.4

13 SQP iterations / 2.67 CPU min. (µVax II)





Constrained Temperature Case ($T \leq 1.45$): could not be solved with sequential method

19

Tricks to generalize classes of problems

Variable Final Time (Miele, 1980)

Define $t = p_{n+1}\, \tau,\ 0 \leq \tau \leq 1,\ p_{n+1} = t_f$

Let $dz/dt = (1/p_{n+1})\, dz/d\tau = f(z, p) \Rightarrow dz/d\tau = (p_{n+1})\, f(z, p)$

Converting Path Constraints to Final Time

Define measure of infeasibility as a new variable, $z_{nz+1}(t)$ (Sargent & Sullivan, 1977):

$$z_{nz+1}(t_f) = \sum_j \int_0^{t_f} \max(0, g_j(z(t), u(t))^2\, dt$$

$$or\ \ \dot{z}_{nz+1}(t) = \sum_j \max(0, g_j(z(t), u(t))^2\ ,\ z_{nz+1}(0) = 0$$

Enforce $z_{nz+1}(t_f) \leq \varepsilon$ (however, constraint is degenerate)

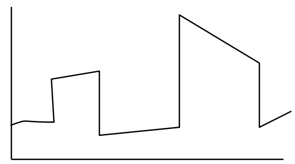
20

Profile Optimization - (Optimal Control)

Optimal Feed Strategy (Schedule) in Batch Reactor

Optimal Startup and Shutdown Policy

Optimal Control of Transients and Upsets



Sequential Approach: Approximate control profile as through parameters (piecewise constant, linear, polynomial, etc.)

Apply NLP to discretization as with parametric optimization

Obtain gradients through adjoints (Hasdorff; Sargent and Sullivan; Goh and Teo) or sensitivity equations (Vassiliadis, Pantelides and Sargent; Gill, Petzold et al.)

Variational (Indirect) approach: Apply optimality conditions and solve as boundary value problem

21

Derivation of Variational Conditions Indirect Approach

<u>Optimality Conditions</u> (Bound constraints on u(t))

$$
\begin{aligned}
&Min \quad \phi(z(t_f)) \\
&s.t. \quad dz/dt = f(z, u),\ z(0) = z_0 \\
&g(z(t_f)) \le 0 \\
&h(z(t_f)) = 0 \\
&a \le u(t) \le b
\end{aligned}
$$

Form Lagrange function - adjoin objective function and constraints:

$$
\begin{aligned}
\phi &= \phi(t_f) + g(z(t_f))^T \mu + h(z(t_f))^T v \\
&+ \int_0^{t_f} \lambda^T (f(z,u) - \dot{z}) + \alpha_a^T (a - u(t)) + \alpha_b^T (u(t) - b)\, dt
\end{aligned}
$$

Integrate by parts :

$$
\begin{aligned}
\phi &= \phi(t_f) + g(z(t_f))^T \mu + h(z(t_f))^T v + \lambda^T(0) z(0) - \lambda^T(t_f) z(t_f) \\
&+ \int_0^{t_f} \dot{\lambda}^T z + \lambda^T f(z,u) + \alpha_a^T (a - u(t)) + \alpha_b^T (u(t) - b)\, dt
\end{aligned}
$$

22

Derivation of Variational Conditions

$$\delta\phi = \left[\frac{\partial \phi}{\partial z} + \frac{\partial g}{\partial z}\mu + \frac{\partial h}{\partial z}v - \lambda\right]^T \delta z(t_f) + \lambda^T(0)\delta z(0)$$

$$+ \int_0^{t_f} \left[\dot{\lambda} + \frac{\partial f(z,u)}{\partial z}\lambda\right]^T \delta z(t) + \left[\frac{\partial f(z,u)}{\partial u}\lambda + \alpha_b - \alpha_a\right]^T \delta u(t)\, dt \geq 0$$

At optimum, $\delta\phi \geq 0$. Since u is the control variable, let all other terms vanish.

$\Rightarrow$ $\delta z(t_f)$: $\lambda(t_f) = \left\{\frac{\partial\phi}{\partial z} + \frac{\partial g}{\partial z}\mu + \frac{\partial h}{\partial z}\gamma\right\}_{t=t_f}$

$\delta z(0)$: $\lambda(0) = 0$ (if z(0) is not specified)

$\delta z(t)$: $\dot{\lambda} = -\frac{\partial H}{\partial z} = -\frac{\partial f}{\partial z}\lambda$

Define Hamiltonian, $H = \lambda^T f(z,u)$

For u <u>not</u> at bound: $\frac{\partial f}{\partial u}\lambda = \frac{\partial H}{\partial u} = 0$

For u <u>at</u> bounds: $\frac{\partial H}{\partial u} = \alpha_a - \alpha_b$

$\alpha_a^T(a - u(t))$

$\alpha_b^T(u(t) - b)$

$u_a \leq u(t) \leq u_b$

$\alpha_a \geq 0, \alpha_b \geq 0$

<u>Upper bound</u>, u(t) = b, $\frac{\partial H}{\partial u} = -\alpha_b \leq 0$

<u>Lower bound</u>, u(t) = a, $\frac{\partial H}{\partial u} = \alpha_a \geq 0$

23

Car Problem

Travel a fixed distance (rest-to-rest) in minimum time.

$$\begin{aligned} &Min \quad t_f \\ &s.t. \;\; x'' = u \\ &a \leq u(t) \leq b \\ &x(0) = 0, x(t_f) = L \\ &x'(0) = 0, x'(t_f) = 0 \end{aligned}$$

$$\begin{aligned} &Min \quad x_3(t_f) \\ &s.t. \;\; x_1' = x_2 \\ &\qquad x_2' = u \\ &\qquad x_3' = 1 \\ &a \leq u(t) \leq b \\ &x_1(0) = 0, x_1(t_f) = L \\ &x_2(0) = 0, x_2(t_f) = 0 \end{aligned}$$

Hamiltonian: $H = \lambda_1 x_2 + \lambda_2 u + \lambda_3$

Adjoints: $\dot{\lambda}_1 = 0 ==> \lambda_1(t) = c_1$

$\dot{\lambda}_2 = -\lambda_1 ==> \lambda_2(t) = c_2 + c_1(t_f - t)$

$\dot{\lambda}_3 = 0 ==> \lambda_3(t_f) = 1,\ \lambda_3(t) = 1$

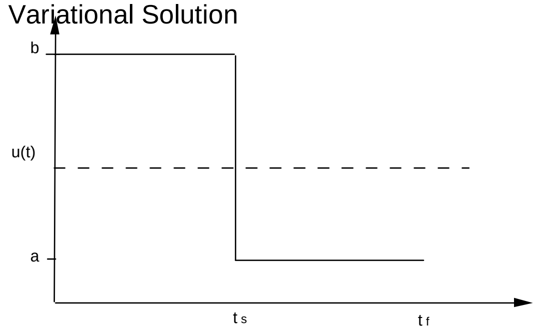
$$\frac{\partial H}{\partial u} = \lambda_2 = c_2 + c_1(t_f - t)\begin{cases} t = 0, c_1 t_f + c_2 < 0, u = b \\ t = t_f, c_2 > 0, u = a \end{cases}$$

Crossover ($\lambda_2 = 0$) occurs at $t = t_s$

24

Car Problem

Analytic Variational Solution



Optimal Profile

From state equations:

$$x_1(t) = \begin{cases} 1/2\ bt^2, & t < t_s \\ 1/2\left(bt_s^2 - a(t_s - t_f)^2\right), & t \ge t_s \end{cases}$$

$$x_2(t) = \begin{cases} bt, & t < t_s \\ bt_s + a(t - t_s), & t \ge t_s \end{cases}$$

Apply boundary conditions at $t = t_f$:

$x_1(t_f) = 1/2\ (b\ t_s^2 - a\ (t_s - t_f)^2) = L$

$x_2(t_f) = bt_s + a\ (t_f - t_s) = 0$

$$\Rightarrow\ t_s = \left[\frac{2L}{b(1 - b/a)}\right]^{1/2}$$

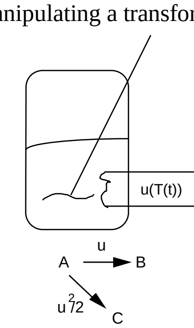
$$t_f = (1 - b/a)\left[\frac{2L}{b(1 - b/a)}\right]^{1/2}$$

- Problem is linear in u(t). Frequently these problems have "bang-bang" character.
- For nonlinear and larger problems, the variational conditions can be solved numerically as boundary value problems.

25

Example: Batch reactor - temperature profile

Maximize yield of B after one hour's operation by manipulating a transformed temperature, *u(t)*.



$$\Rightarrow \quad Min\ \ -b(1)$$
$$s.t.\quad a' = -(u + u^2/2)a,\ a(0) = 1$$
$$b' = ua,\ b(0) = 0$$
$$0 \le u(t) \le 5$$

Optimality conditions:

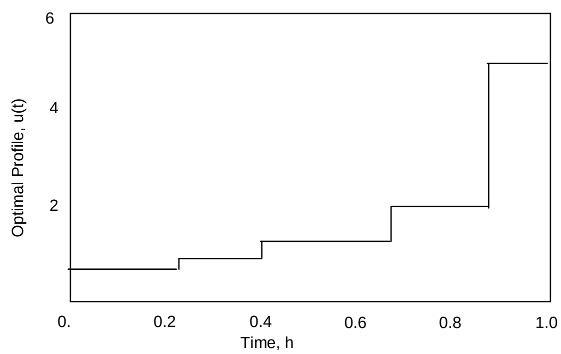
$$H = -\lambda_a(u + u^2/2)a + \lambda_b ua$$
$$\partial H/\partial u = -\lambda_a(1 + u)a + \lambda_b a = \alpha_0 - \alpha_5$$
$$0 \le \alpha_0 \perp u \ge 0,\ \ 0 \le \alpha_5 \perp (5 - u) \ge 0$$
$$\lambda_a' = \lambda_a(u + u^2/2) - \lambda_b u,\ \lambda_a(0) = 0$$
$$\lambda_b' = 0,\ \lambda_b(0) = -1$$

Cases Considered

1. NLP Approach - piecewise constant and linear profiles.
2. Indirect Approach

26

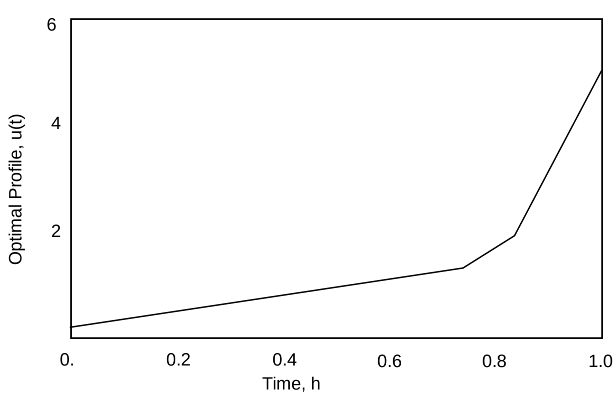
Batch Reactor Optimal Temperature Program
Piecewise Constant



Results
Piecewise Constant Approximation with Variable Time Elements
Optimum B/A: 0.57105

27

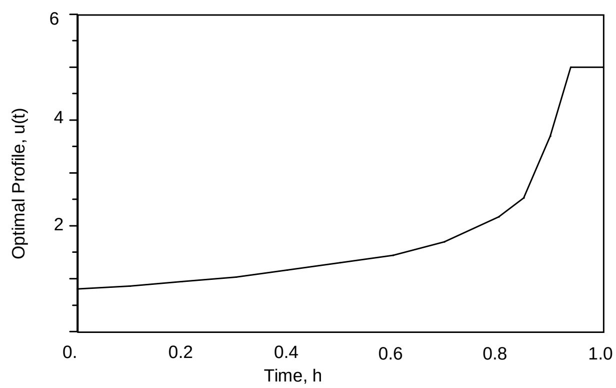
Batch Reactor Optimal Temperature Program
Piecewise Linear



Results:
Piecewise Linear Approximation with Variable Time Elements
Optimum B/A: 0.5726
Equivalent # of ODE solutions: 32

28

Batch Reactor Optimal Temperature Program
Indirect Approach



Results:
Control Vector Iteration with Conjugate Gradients
Optimum (B/A): 0.5732
Equivalent # of ODE solutions: 58

29

Dynamic Optimization - Sequential Strategies

Small NLP problem, $O(n_p+n_u)$ (large-scale NLP solver not required)

- Use NPSOL, NLPQL, etc.
- Second derivatives difficult to get

Repeated solution of DAE model and sensitivity/adjoint equations, scales with n_z and n_p

- Dominant computational cost
- May fail at intermediate points

Sequential optimization is not recommended for unstable systems. State variables blow up at intermediate iterations for control variables and parameters.

Discretize control profiles to parameters (at what level?)

Path constraints are difficult to handle exactly for NLP approach

30

Instabilities in DAE Models

This example cannot be solved with sequential methods (Bock, 1983):

$$dy_1/dt = y_2$$

$$dy_2/dt = \tau^2\, y_1 + (\pi^2 - \tau^2) \sin(\pi\, t)$$

The characteristic solution to these equations is given by:

$$y_1(t) = \sin(\pi\, t) + c_1 \exp(-\tau\, t) + c_2 \exp(\tau\, t)$$

$$y_2\,(t) = \pi \cos(\pi\, t) - c_1\, \tau \exp(-\,\tau\, t) + c_2\, \tau \exp(\tau\, t)$$

Both c_1 and c_2 can be set to zero by either of the following equivalent conditions:

IVP $\quad y_1(0) = 0,\ y_2\,(0) = \pi$

BVP $\quad y_1(0) = 0,\ y_1(1) = 0$

31

IVP Solution

If we now add roundoff errors e_1 and e_2 to the IVP and BVP conditions, we see significant differences in the sensitivities of the solutions.

For the IVP case, the sensitivity to the *analytic* solution profile is seen by large changes in the profiles $y_1(t)$ and $y_2(t)$ given by:

$$y_1(t) = \sin(\pi\, t) + (e_1 - e_2/\tau) \exp(-\tau\, t)/2$$
$$+(e_1 + e_2/\tau) \exp(\tau\, t)/2$$

$$y_2\,(t) = \pi \cos(\pi\, t) - (\tau\, e_1 - e_2) \exp(-\tau\, t)/2$$
$$+ (\tau\, e_1 + e_2) \exp(\tau\, t)/2$$

Therefore, even if e_1 and e_2 are at the level of machine precision ($< 10^{-13}$), a large value of τ and t will lead to unbounded solution profiles.

32

BVP Solution

On the other hand, for the boundary value problem, the errors affect the *analytic* solution profiles in the following way:

$$y_1(t) = \sin(\pi t) + [e_1 \exp(\tau) - e_2] \exp(-\tau t)/[\exp(\tau) - \exp(-\tau)]$$

$$+ [e_1 \exp(-\tau) - e_2] \exp(\tau t)/[\exp(\tau) - \exp(-\tau)]$$

$$y_2(t) = \pi \cos(\pi t) - \tau [e_1 \exp(\tau) - e_2] \exp(-\tau t)/[\exp(\tau) - \exp(-\tau)]$$

$$+ \tau [e_1 \exp(-\tau) - e_2] \exp(\tau t)/[\exp(\tau) - \exp(-\tau)]$$

Errors in these profiles never exceed $\tau (e_1 + e_2)$, and as a result a solution to the BVP is readily obtained.

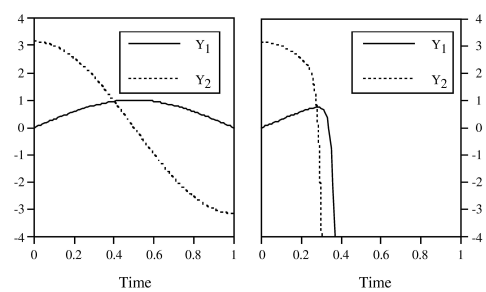
33

BVP and IVP Profiles

$e_1, e_2 = 10^{-9}$

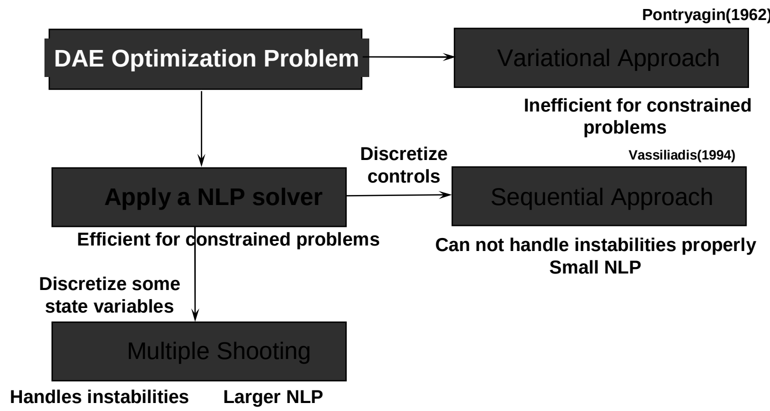
Linear BVP solves easily

IVP blows up before midpoint



34

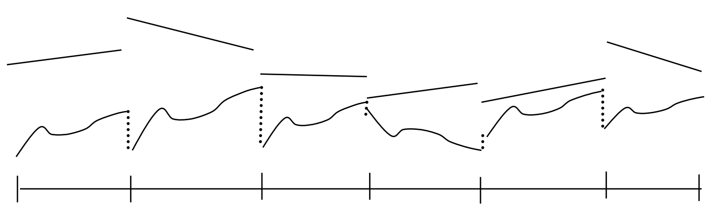
Dynamic Optimization Approaches



35

Multiple Shooting for Dynamic Optimization

Divide time domain into separate regions



Integrate DAEs state equations over each region

Evaluate sensitivities in each region as in sequential approach wrt u_{ij}, p and z_j

Impose matching constraints in NLP for state variables over each region

Variables in NLP are due to control profiles as well as initial conditions in each region

36

Multiple Shooting
Nonlinear Programming Problem

$$\min_{u_{i,j},\,p} \ \psi\big(z(t_f),\, y(t_f)\big)$$

s.t.

$$z(z_j, u_{i,j}, p, t_{j+1}) - z_{j+1} = 0$$

$$z_k^{\ l} \le z(z_j, u_{i,j}, p, t_k) \le z_k^{\ u}$$

$$y_k^{\ l} \le y(z_j, u_{i,j}, p, t_k) \le y_k^{\ u}$$

$$u_i^{\ l} \le u_{i,j} \le u_i^{\ u}$$

$$p^l \le p \le p^u$$

$$\left(\frac{dz}{dt}\right) = f\big(z, y, u_{i,j}, p\big), \quad z(t_j) = z_j$$

$$g\big(z, y, u_{i,j}, p\big) = 0$$

$$z_0^{\ o} = z\,(0)$$

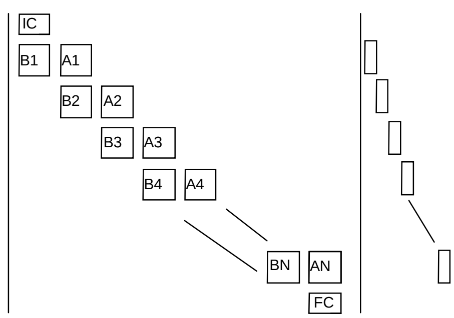
Solved Implicitly

$$\min_{x \in \Re^n} f(x)$$

$$\text{s.t} \quad c(x) = 0$$

$$x^L \le x \le x^u$$

BVP Problem Decomposition



Consider: Jacobian of Constraint Matrix for NLP

- bound unstable modes with boundary conditions (dichotomy)
- can be done implicitly by determining stable pivot sequences in multiple shooting constraints approach
- well-conditioned problem implies dichotomy in BVP problem (deHoog and Mattheij)

Bock Problem (with $\tau = 50$)

- Sequential approach blows up (starting within 10^{-9} of optimum)
- Multiple Shooting optimization requires 4 SQP iterations

Dynamic Optimization – Multiple Shooting Strategies

Larger NLP problem $O(n_p+NE\ (n_u+n_z))$

- Use SNOPT, MINOS, etc.
- Second derivatives difficult to get

Repeated solution of DAE model and sensitivity/adjoint equations, scales with nz and np

- Dominant computational cost
- May fail at intermediate points

Multiple shooting can deal with unstable systems with sufficient time elements.

Discretize control profiles to parameters (at what level?)

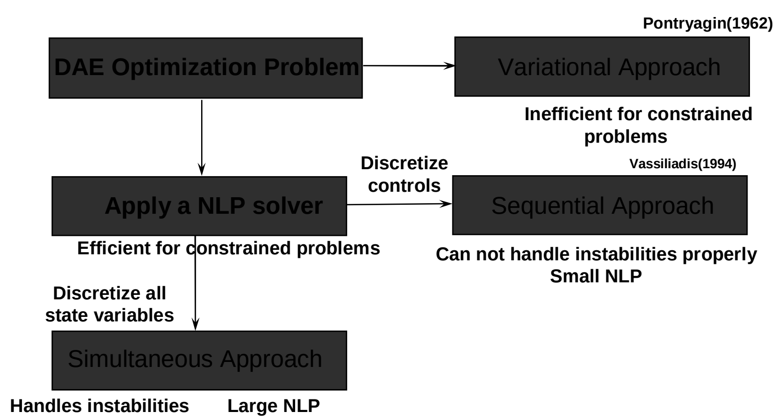
Path constraints are difficult to handle exactly for NLP approach

Block elements for each element are dense!

Extensive developments and applications by Bock and coworkers using MUSCOD code

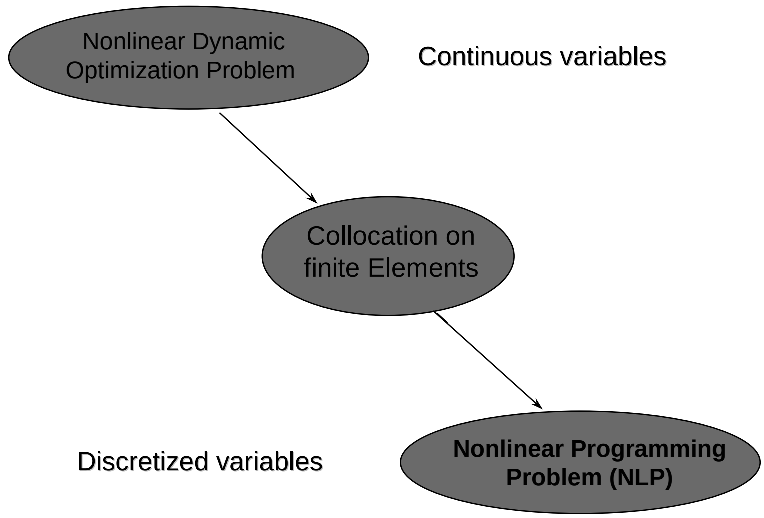
39

Dynamic Optimization Approaches



40

Nonlinear Programming Formulation



41

Discretization of Differential Equations Orthogonal Collocation

<u>Given:</u> *dz/dt = f(z, u, p), z(0)=given*

<u>Approximate</u> *z* and *u* by Lagrange interpolation polynomials (order K+1 and K, respectively) with interpolation points, t_k

$$z_{K+1}(t) = \sum_{k=0}^{K} z_k \ell_k(t), \ell_k(t) = \prod_{\substack{j=0 \\ j \neq k}}^{K} \frac{(t-t_j)}{(t_k - t_j)} \Longrightarrow z_{K+1}(t_k) = z_k$$

$$u_K(t) = \sum_{k=1}^{K} u_k \ell_k(t), \ell_k(t) = \prod_{\substack{j=1 \\ j \neq k}}^{K} \frac{(t-t_j)}{(t_k - t_j)} \Longrightarrow u_K(t_k) = u_k$$

Substitute z_{N+1} and u_N into ODE and apply equations at t_k.

$$r(t_k) = \sum_{j=0}^{K} z_j \dot{\ell}_j(t_k) - f(z_k, u_k) = 0, \quad k = 1,...K$$

42

Collocation Example

$$z_{K+1}(t) = \sum_{k=0}^{K} z_k \ell_k(t), \ell_k(t) = \prod_{\substack{j=0 \\ j \neq k}}^{K} \frac{(t - t_j)}{(t_k - t_j)} \Longrightarrow z_{N+1}(t_k) = z_k$$

$t_0 = 0, t_1 = 0.21132, t_2 = 0.78868$

$\ell_0(t) = 6t^2 - 6t + 1, \quad \dot{\ell}_0(t) = 12t - 6$

$\ell_1(t) = -8.195t^2 + 6.4483t, \quad \dot{\ell}_1(t) = 6.4483 - 16.39t$

$\ell_2(t) = 2.19625t^2 - 0.4641t, \quad \dot{\ell}_2(t) = 4.392t - 0.4641$

Solve $z' = z^2 - 3z + 2, z(0) = 0$

$\Longrightarrow z_0 = 0$

$z_0 \dot{\ell}_0(t_1) + z_1 \dot{\ell}_1(t_1) + z_2 \dot{\ell}_2(t_1) = z_1^2 - 3z_1 + 2$

$(2.9857 z_1 + 0.46412 z_2 = z_1^2 - 3z_1 + 2)$

$z_0 \dot{\ell}_0(t_2) + z_1 \dot{\ell}_1(t_2) + z_2 \dot{\ell}_2(t_2) = z_2^2 - 3z_2 + 2$

$(-6.478 z_1 + 3z_2 = z_2^2 - 3z_2 + 2)$

$z_0 = 0, z_1 = 0.291 \ (0.319), z_2 = 0.7384 \ (0.706)$

$z(t) = 1.5337t - 0.76303t^2$

43

Converted Optimal Control Problem Using Collocation

Min $\phi(z(t_f))$

s.t. $z' = f(z, u, p), z(0)=z_0$

$g(z(t), u(t), p) \leq 0$

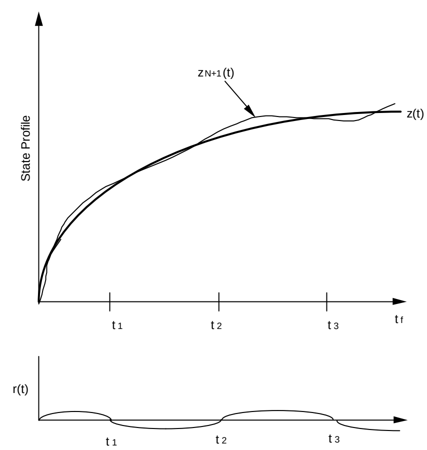
$h(z(t), u(t), p) = 0$

to Nonlinear Program

Min $\phi(z_f)$

$$\left.\begin{aligned} &\sum_{j=0}^{K} z_j \dot{\ell}_j(t_k) - f(z_k, u_k) = 0, \ z_0 = z(0) \\ &g(z_k, u_k) \leq 0 \\ &h(z_k, u_k) = 0 \end{aligned}\right\} k = 1, \ldots K$$

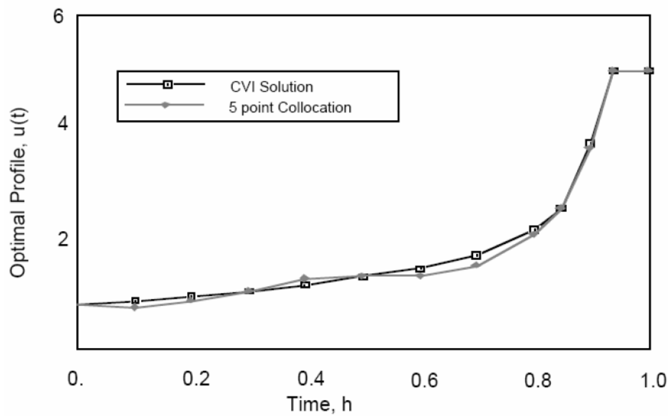
$$\sum_{j=0}^{K} z_j \ell_j(1) - z_f = 0$$



How accurate is approximation

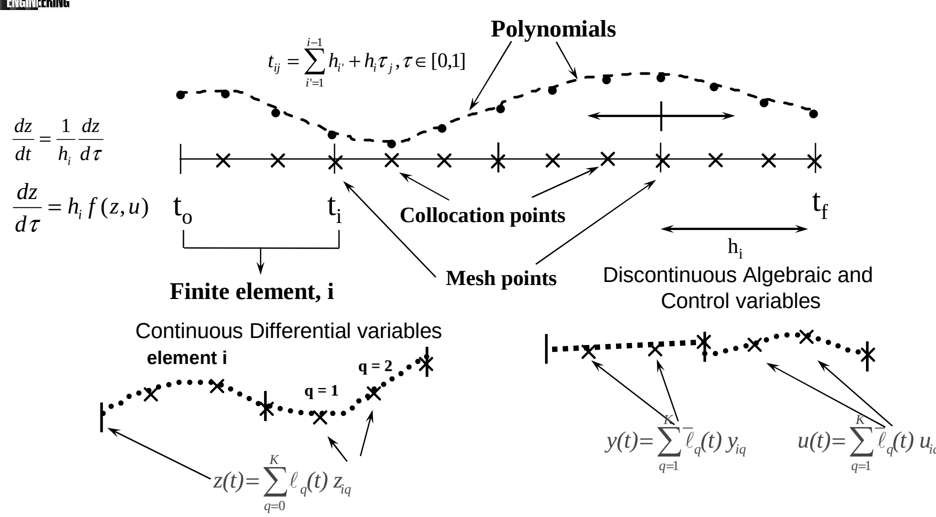
44

Results of Optimal Temperature Program
Batch Reactor (Revisited)



Results - NLP with Orthogonal Collocation
Optimum B/A - 0.5728
# of ODE Solutions - 0.7   (Equivalent)

Collocation on Finite Elements



$$r(t_{ik}) = \sum_{j=0}^{K} (z_{ij}\dot{\ell}_j(\tau_k)) - h_i f(z_{ik}, u_{ik}, p) = 0, \quad k = 1,..K, i = 1,..NE$$

Nonlinear Programming Problem

$$\min \ \psi(z_f)$$

s.t.

$$\sum_{j=0}^{K}(z_{ij}\dot{\ell}_j(\tau_k)) - h_i f(z_{ik}, u_{ik}, p) = 0$$

$$g(z_{i,k}, y_{i,k}, u_{i,k}, p) = 0$$

$$\sum_{j=0}^{K}(z_{i-1,j}\ell_j(1)) - z_{i0} = 0, \ i = 2,..NE$$

$$\sum_{j=0}^{K}(z_{NE,j}\ell_j(1)) - z_f = 0, \ z_{10} = z(0)$$

$$z_{i,j}^{\ l} \le z_{i,j} \le z_{i,j}^{\ u}$$

$$y_{i,j}^{\ l} \le y_{i,j} \le y_{i,j}^{\ u}$$

$$u_{i,j}^{\ l} \le u_{i,j} \le u_{i,j}^{\ u}$$

$$p^l \le p \le p^u$$

$$\min_{x \in \Re^n} f(x)$$

$$\text{s.t} \quad c(x) = 0$$

$$x^L \le x \le x^u$$

Finite elements, h_i, can also be variable to determine break points for *u(t).*

Add $h_u \ge h_i \ge 0$, $\Sigma h_i = t_f$

Can add constraints $g(h, z, u) \le \varepsilon$ for approximation error

47

Hot Spot Reactor Revisited

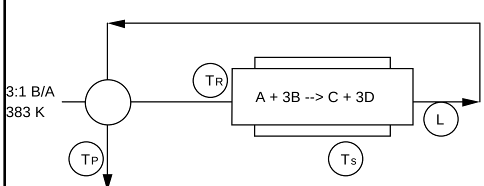
$$\underset{T_P, T_R, L, T_S}{Min} \quad \Phi = L - \int_0^L (T(t) - T_S / T_R)\,dt$$

$$s.t. \quad \frac{dq}{dt} = 0.3(1 - q(t))\exp[20 - 20/T(t)], \ q(0) = 0$$

$$\frac{dT}{dt} = -1.5(T(t) - T_S/T_R) + 2/3\frac{dq}{dt}, \ T(0) = 1$$

$$\Delta H_{feed}(T_R, 110^o C) - \Delta H_{product}(T_P, T(L)) = 0$$

$$T_P = 120^o C, \ T(L) = 1 + 10^o C/T_R$$



T_P = specified product temperature
T_R = reactor inlet, reference temperature
L = reactor length
T_S = steam sink temperature
q(t) = reactor conversion profile
T(t) = normalized reactor temperature profile

Cases considered:

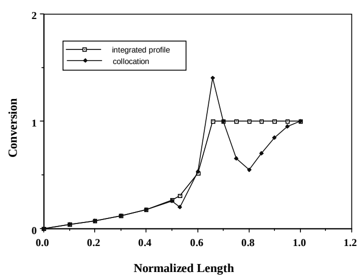
- Hot Spot - no state variable constraints
- Hot Spot with $T(t) \le 1.45$

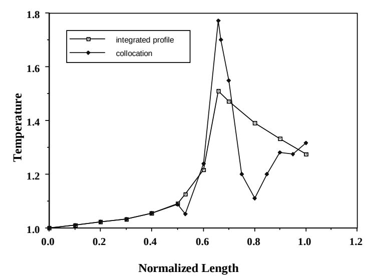
48

Base Case Simulation

Method: OCFE at initial point with 6 equally spaced elements

	L(norm)	T_R(K)	T_S(K)	T_P(K)
Base Case:	1.0	462.23	425.26	250





49

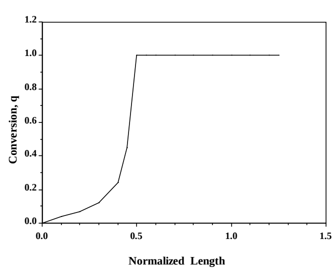
Unconstrained Case

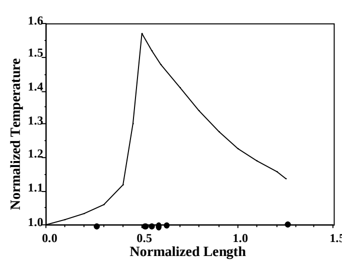
Method: OCFE combined formulation with rSQP
identical to integrated profiles at optimum

	L(norm)	T_R(K)	T_S(K)	T_P(K)
Initial:	1.0	462.23	425.26	250
Optimal:	1.25	500	470.1	188.4

123 CPU s. (μVax II)

$\phi^* = -171.5$





50

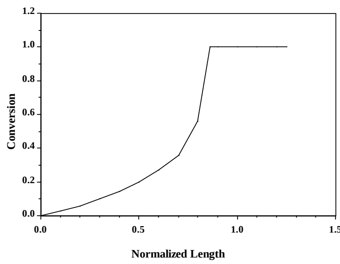
Temperature Constrained Case

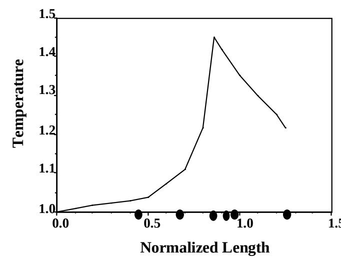
T(t) ≤ 1.45

Method: OCFE combined formulation with rSQP, identical to integrated profiles at optimum

	L(norm)	T_R(K)	T_S(K)	T_P(K)
Initial:	1.0	462.23	425.26	250
Optimal:	1.25	500	450.5	232.1

57 CPU s. (μVax II), $\phi^* = -148.5$





Theoretical Properties of Simultaneous Method

A. <u>Stability and Accuracy of Orthogonal Collocation</u>

- Equivalent to performing a *fully implicit* Runge-Kutta integration of the DAE models at Gaussian (Radau) points
- 2K order (2K-1) method which uses K collocation points
- Algebraically stable (i.e., possesses A, B, AN and BN stability)

B. <u>Analysis of the Optimality Conditions</u>

- An equivalence has been established between the Kuhn-Tucker conditions of NLP and the variational necessary conditions
- Rates of convergence have been established for the NLP method

Dynamic Optimization Engines

Evolution of NLP Solvers:

➔ ***for dynamic optimization, control and estimation***

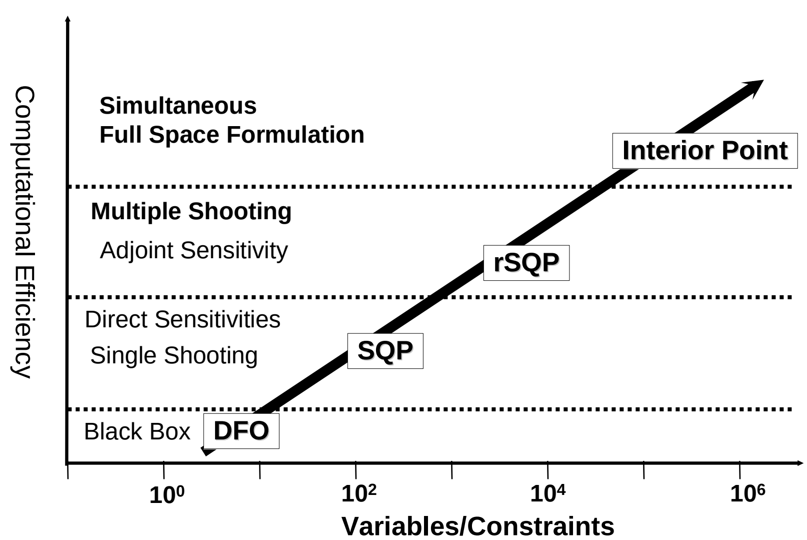
SQP → rSQP → Full-space Barrier

E.g., ***IPOPT*** - Simultaneous dynamic optimization over 1 000 000 variables and constraints

Object Oriented Codes tailored to structure, sparse linear algebra and computer architecture (e.g., IPOPT 3.2)

53

Hierarchy of Nonlinear Programming for Dynamic Optimization Formulations



54

Comparison of Computational Complexity

($\alpha \in [2, 3]$, $\beta \in [1, 2]$, n_w, n_u - assume $N_m = O(N)$)

	Single Shooting	Multiple Shooting	Simultaneous
DAE Integration	$n_w^{\beta} N$	$n_w^{\beta} N$	---
Sensitivity	$(n_w N)(n_u N)$	$(n_w N)(n_u + n_w)$	$N(n_u + n_w)$
Exact Hessian	$(n_w N)(n_u N)^2$	$(n_w N)(n_u + n_w)^2$	$N(n_u + n_w)$
NLP Decomposition	---	$n_w^3 N$	---
Step Determination	$(n_u N)^{\alpha}$	$(n_u N)^{\alpha}$	$((n_u + n_w)N)^{\beta}$
Backsolve	---	---	$((n_u + n_w)N)$
	$O((n_u N)^{\alpha} + N^2 n_w n_u + N^3 n_w n_u^2)$	$O((n_u N)^{\alpha} + N n_w^3 + N n_w (n_w + n_u)^2)$	$O((n_u + n_w)N)^{\beta}$

55

Simultaneous DAE Optimization

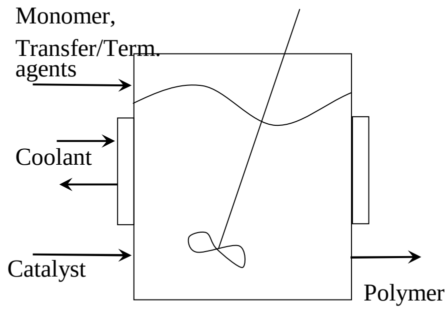
Case Studies

- Reactor - Based Flowsheets
- Fed-Batch Penicillin Fermenter
- Temperature Profiles for Batch Reactors
- Parameter Estimation of Batch Data
- Synthesis of Reactor Networks
- Batch Crystallization Temperature Profiles
- Grade Transition for LDPE Process
- Ramping for Continuous Columns
- Reflux Profiles for Batch Distillation and Column Design
- Source Detection for Municipal Water Networks
- Air Traffic Conflict Resolution
- Satellite Trajectories in Astronautics
- Batch Process Integration
- Optimization of Simulated Moving Beds

56

Production of High Impact Polystyrene (HIPS)

Startup and Transition Policies (Flores et al., 2005a)



Initiation reactions

Thermal

$$3M_S \xrightarrow{k_{i0}} 2R_S^1$$

Chemical

$$I \xrightarrow{f_1 k_d} 2R$$
$$R + M_S \xrightarrow{k_{i1}} R_S^1$$
$$R + B_0 \xrightarrow{k_{i2}} B_R$$
$$B_R + M_S \xrightarrow{k_{i3}} B_{RS}^1$$

Propagation reactions

$$R_S^j + M_S \xrightarrow{k_p} R_S^{j+1}$$
$$B_{RS}^j + M_S \xrightarrow{k_p} B_{RS}^{j+1}$$

Definite termination reactions

Homopolymer

$$R_S^j + R_S^m \xrightarrow{k_t} P^{j+m}$$

Grafting

$$R_S^j + B_R \xrightarrow{k_t} B_P^j$$
$$R_S^j + B_{RS}^m \xrightarrow{k_t} B_P^{j+m}$$

Crosslinking

$$B_R + B_R \xrightarrow{k_t} B_{EB}$$
$$B_{RS}^j + B_R \xrightarrow{k_t} B_{PB}^j$$
$$B_{RS}^j + B_{RS}^m \xrightarrow{k_t} B_{PB}^{j+m}$$

Transfer reactions

Monomer

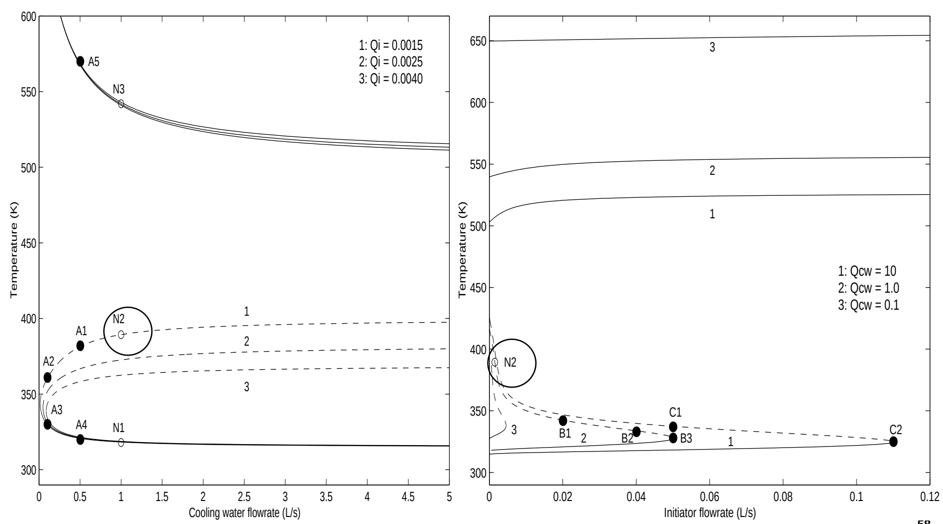
$$R_S^j + M_S \xrightarrow{k_{fs}} P^j + R_S^1$$
$$B_{RS}^j + M_S \xrightarrow{k_{fs}} B_P^j + R_S^1$$

Grafting sites

$$R_S^j + B_0 \xrightarrow{k_{fb}} P^j + B_R$$
$$B_{RS}^j + B_0 \xrightarrow{k_{fb}} B_P^j + B_R$$

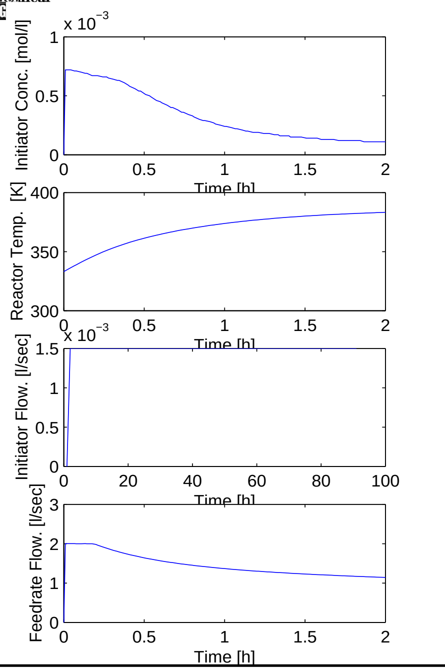
Phase Diagram of Steady States

Transitions considered among all steady state pairs



58

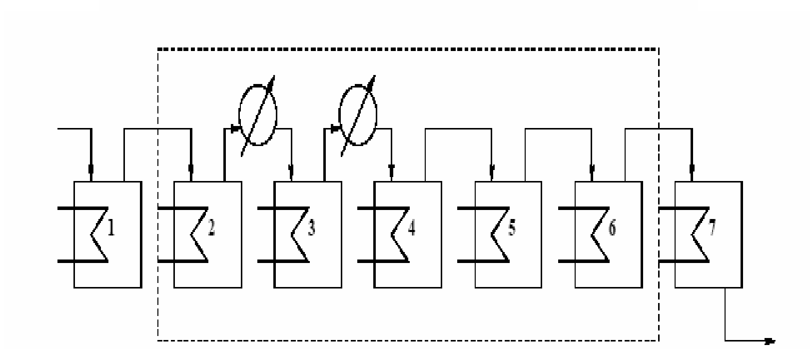
Startup to Unstable Steady State



- 926 variables
- 476 constraints
- 36 iters. / 0.95 CPU s (P4)

59

HIPS Process Plant (Flores et al., 2005b)



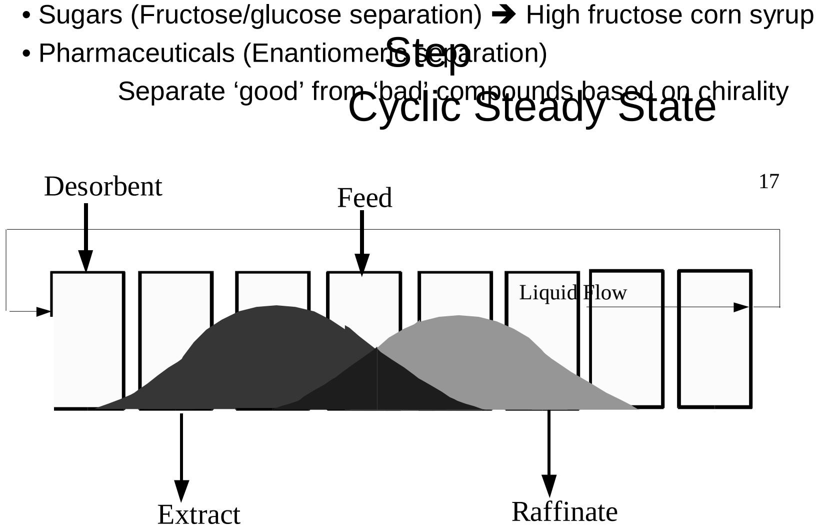
- Many grade transitions considered with stable/unstable pairs
- 1-6 CPU min (P4) with IPOPT
- Study shows benefit for sequence of grade changes to achieve wide range of grade transitions.

60

SMB Applications

- Petrochemical (Xylene isomers)
- Sugars (Fructose/glucose separation) ➔ High fructose corn syrup
- Pharmaceuticals (Enantiomers separation)

  Separate ‘good’ from ‘bad’ compounds based on chirality



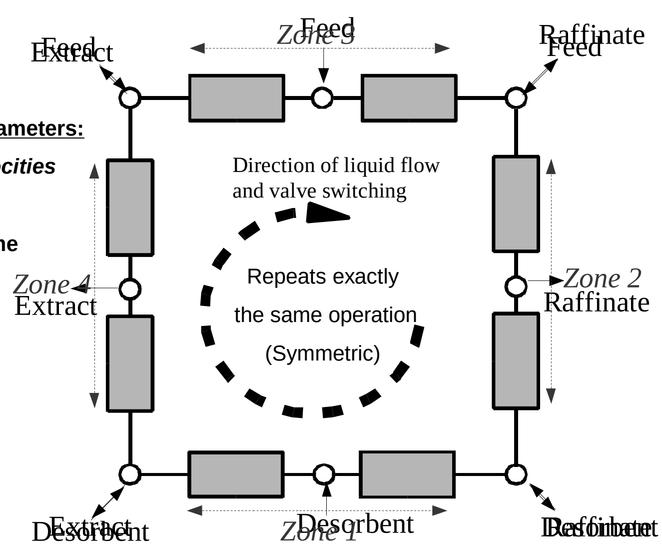
Simulated Moving Bed

Operating parameters:

4 Zone velocities

+

Step time



Formulation of Optimization Problem

Zone velocities Step time

$$\max_{\boxed{u_I(t), u_{II}(t), u_{III}(t), u_{IV}(t)}, \boxed{t_{step}}} \quad \bar{u}_F := \frac{1}{t_{step}} \int_0^{t_{step}} u_F(t)dt$$

(Maximize average feed velocity)

Product requirements

$$(Extract \quad Product \quad Purity) = \frac{\int_0^{t_{step}} u_E(t) C_{E,k}(t)dt}{\sum_{i=1}^{Nc} \int_0^{t_{step}} u_E(t) C_{E,i}(t)dt} \geq Pur_{min}$$

$$(Extract \quad Product \quad Recovery) = \frac{\int_0^{t_{step}} u_E(t) C_{E,k}(t)dt}{\int_0^{t_{step}} u_F(t) C_{F,k}(t)dt} \geq Rec_{min}$$

Bounds on liquid velocities

$$u_l \leq u_m(t) \leq u_u \qquad m = I, II, III, IV$$

SMB model

$$\epsilon \frac{\partial C_i(x,t)}{\partial t} + (1-\epsilon)\frac{\partial q_i(x,t)}{\partial t} + u_m \frac{\partial C_i(x,t)}{\partial x} = 0$$

$$(1-\epsilon)\frac{\partial q_i(x,t)}{\partial t} = K_{app\,i}(C_i(x,t) - C_i^{eq}(x,t))$$

$$q_{n,i}(x,t) = K_i C_{n,i}^{eq}(x,t)$$

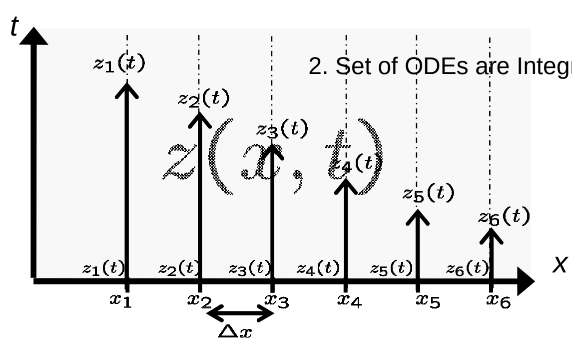
CSS constraint

$$C_i(x,t) = C_i(x, t + t_{step})$$

$$q_i(x,t) = q_i(x, t + t_{step})$$

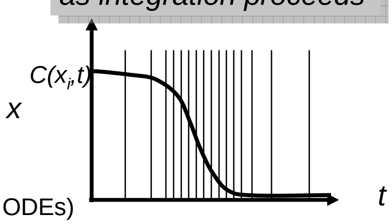
63

Treatment of PDEs: Single Discretization



2. Set of ODEs are Integrated

Step size determined as integration proceeds



1. PDE is discretized only in *x* (turn a PDE into ODEs)

$$\min_p \Phi\left(z(x,t_f)\right)$$

$$subject \quad to:$$

$$f_i\left(p, \left(\frac{\Delta z}{\Delta x}\right)_i, \frac{dz_i(t)}{dt}\right) = 0, \quad \left(\frac{\Delta z}{\Delta x}\right)_i = \frac{z_{i+1}(t) - z_{i-1}(t)}{x_{i+1} - x_{i-1}}$$

$$g\left(p, z(x,t)\right) \leq 0$$

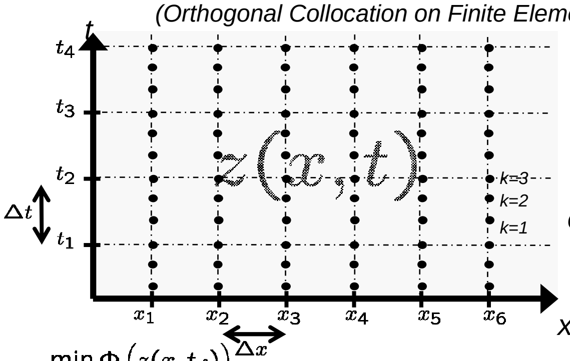
$$h\left(p, z(x,t)\right) = 0$$

PDE $\rightarrow$ *ODE (Handled by integrator)*

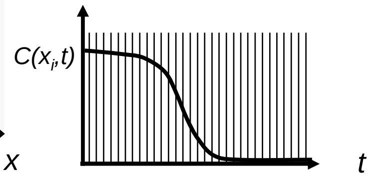
64

Treatment of PDEs: Simultaneous Approach

(Orthogonal Collocation on Finite Elements)



Step size is determined a priori



Huge number of variables (handled by optimizer)

$$\min_p \Phi\left(z(x,t_f)\right)$$

subject to :

$$f_{i,j,k}\left(p,\left(\frac{\Delta z}{\Delta x}\right)_{i,j,k},\left(\frac{dz}{dt}\right)_{i,j,k}\right)=0,\quad \left(\frac{\Delta z}{\Delta x}\right)_{i,j,k}=\frac{z_{i+1,j,k}-z_{i-1,j,k}}{x_{i+1,j}-x_{i-1,j}}$$

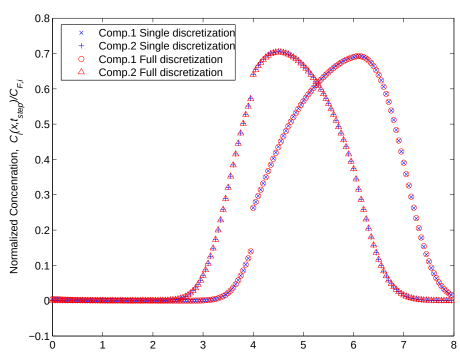
$$g\left(p,z(x,t)\right)\leq 0 \qquad z_{i,j,k}=z_{i,j}^{0}+h_j\sum_{k=1}^{3}\Omega_{j,k}\left(\frac{dz}{dt}\right)_{i,j,k}$$

$$h\left(p,z(x,t)\right)=0$$

PDE → Algebraic equations

65

Comparison of two approaches



(Linear isotherm, fructose/glucose separation)

Shooting and Simultaneous methods find the same optimal solution

Initial feed velocity: 0.01 m/h

Optimization

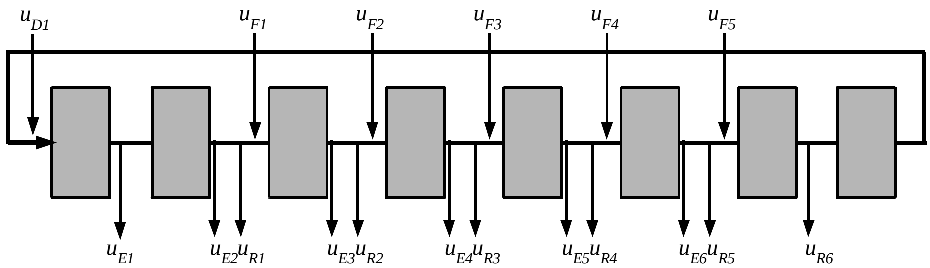
Optimal feed velocity: 0.52 m/h

**On Pentium IV 2.8GHz*

	# of variables	CPU Time*	# of iteration
Shooting Approach *Implemented on gPROMS, solved using SRQPD*	644	*111.8 min* *(89% spent by integrator)*	49
Simultaneous Approach *Implemented on AMPL, solved using IPOPT*	33999	**1.53 min**	47

66

Superstructure for Zone Configuration



Constraints:

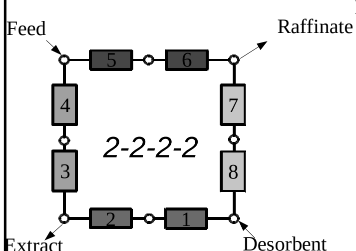
All velocities are constant

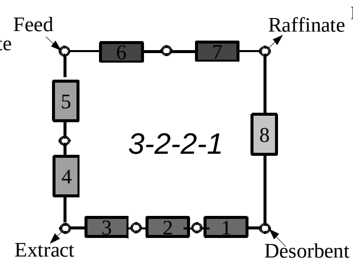
8 columns, multiple streams

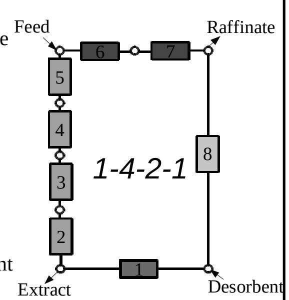
Repeat exactly same stream policies for each step

67

Allows straightforward zone configuration optimization







Optimal feed velocity: 0.906 m/h

0.957 m/h

1.052 m/h

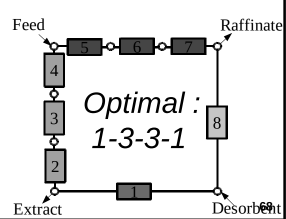
# of columns	*8*	10	12	14
# of zone configurations	*35*	84	165	286

Too many configurations to enumerate!

*Superstructure configuration **AVOIDS** enumerating all configurations*

CPU Time: 1.73 min*

Optimal feed velocity: 1.158 m/h

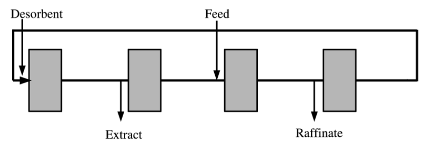


**On Xeon 3.2GHz*

68

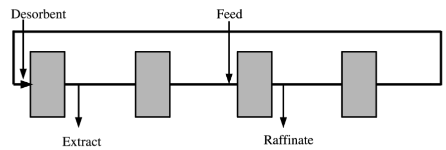
Nonstandard SMB: Addressed by Extended Superstructure NLP

- **Standard SMB**



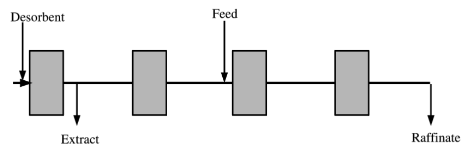
- **VARICOL**

(Asynchronous switching)

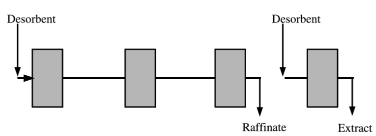


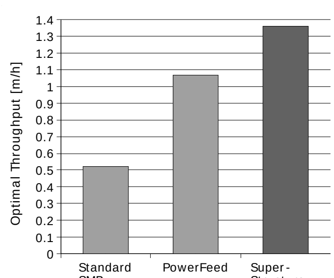
- **Three Zone**

(Circulation loop is cut open)



Optimal Operating Scheme:
Result of Superstructure Optimization

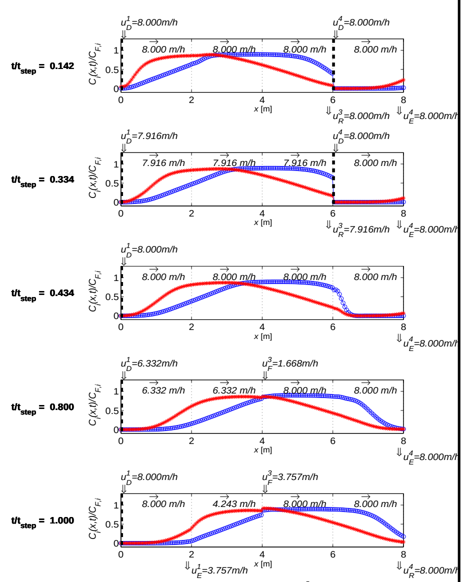




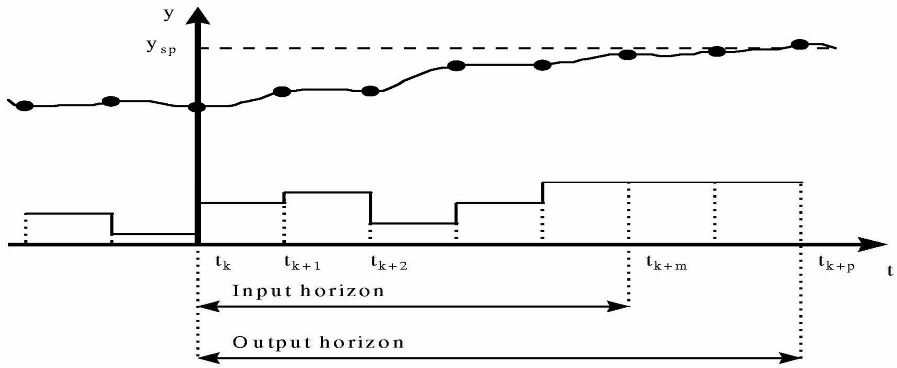
CPU Time for optimization: 9.03 min*

34098 variables, 34013 equations

on Xeon 3.2 GHz



Nonlinear Model Predictive Control (NMPC)



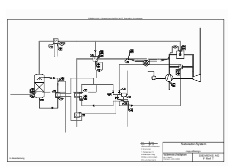
$$\min_{\boldsymbol{u}} \quad \sum \| y(t) - y^{\text{sp}} \|_{Q^y} + \sum \| \boldsymbol{u}(\boldsymbol{t}^k) - \boldsymbol{u}(\boldsymbol{t}^{k-1}) \|_{Q^u}$$

$$\begin{aligned} s.t. \quad & z'(t) = F(z(t), y(t), \boldsymbol{u}(t), t) \\ & 0 = G(z(t), y(t), \boldsymbol{u}(t), t) \\ & z(t) = z^{\text{init}} \\ & \text{Bound Constraints} \\ & \text{Other Constraints} \end{aligned}$$

71

Dynamic optimization in a MATLAB Framework

Dynamic Process



Dynamic Optimization Problem

Process Model

$\mathbf{f}(\mathbf{x}', \mathbf{x}, \mathbf{y}, \mathbf{u}, \mathbf{p}, t) = \mathbf{0}$

$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{p}, t) = \mathbf{0}$

Inequality Constraints

$\mathbf{h}(\mathbf{x}', \mathbf{x}, \mathbf{y}, \mathbf{u}, \mathbf{p}, t) \le 0$

Initial Conditions

$\boldsymbol{x}(t_0) = \boldsymbol{x_0}$

Constraints at Final Time

$\varphi(\mathbf{x}'(t_f), \mathbf{x}(t_f), \mathbf{y}(t_f), \mathbf{u}(t_f), \mathbf{x}_0, \mathbf{p}, t_f) = 0$

Objective Function

$\min_{\mathbf{u}(t), \mathbf{p}, \mathbf{x}_0, t_f} \mathrm{P}(\mathbf{x}(t_f), \mathbf{y}(t_f), \mathbf{u}(t_f), \mathbf{p}, \mathbf{x}_0, t_f)$

Discretization Method
No. of Time Elements
Collocation Order

Full Discretization of State and Control Variables

NLP Optimization Problem

Process Model

$\hat{\mathbf{f}}(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{u}}, \mathbf{p}, t_f, \mathbf{x_0}) = \mathbf{0}$

$\hat{\mathbf{g}}(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{u}}, \mathbf{p}, t_f) = \mathbf{0}$

Inequality Constraints

$\hat{\mathbf{h}}(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{u}}, \mathbf{p}, t_f) \le 0$

Constraints at Final Time

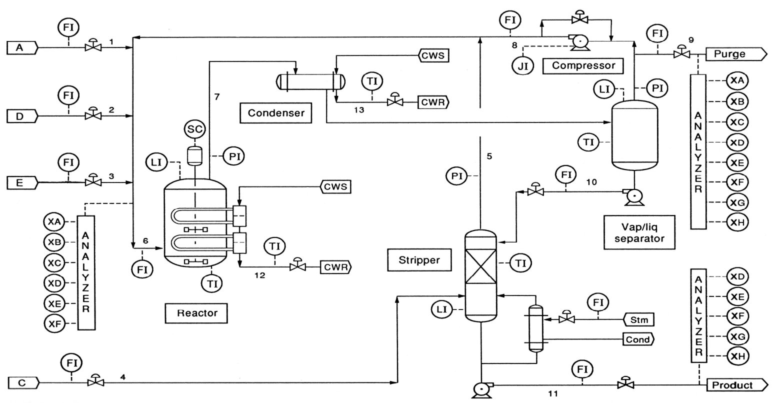
$\varphi(\mathbf{x}_{N_t}, \mathbf{y}_{N_t}, \mathbf{u}_{N_t}, \mathbf{p}, t_f) = 0$

Objective Function

$\min \mathrm{P}(\mathbf{x}_{N_t}, \mathbf{y}_{N_t}, \mathbf{u}_{N_t}, \mathbf{p}, t_f)$

72

Tennessee Eastman Process



Unstable Reactor

11 Controls; Product, Purge streams

Model extended with energy balances

Tennessee Eastman Challenge Process

DAE Model	
Number of differential equations	30
Number of algebraic variables	152
Number of algebraic equations	141
Difference (control variables)	11

NLP Optimization problem	
Number of variables of which are fixed	10920 0
Number of constraints	10260
Number of lower bounds	780
Number of upper bounds	540
Number of nonzeros in Jacobian	49230
Number of nonzeros in Hessian	14700

Method of Full Discretization of State and Control Variables

Large-scale Sparse block-diagonal NLP

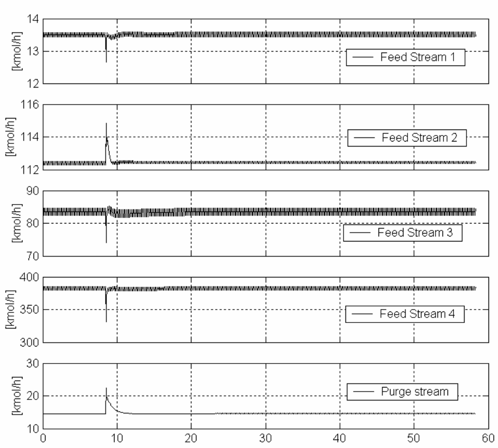
Setpoint change studies

Process variable	Type	Magnitude
Production rate change	Step	-15% Make a step change to the variable(s) used to set the process production rate so that the product flow leaving the stripper column base changes from 14,228 to 12,094 kg h^{-1}
Reactor operating pressure change	Step	-60 kPa Make a step change so that the reactor operating pressure changes from 2805 to 2745 kPa
Purge gas composition of component B change	Step	+2% Make a step change so that the composition of component B in the gas purge changes from 13.82 to 15.82%

Setpoint changes for the base case [Downs & Vogel]

75

Case Study:
Change Reactor pressure by 60 kPa



Control profiles

All profiles return to their base case values

Same production rate

Same product quality

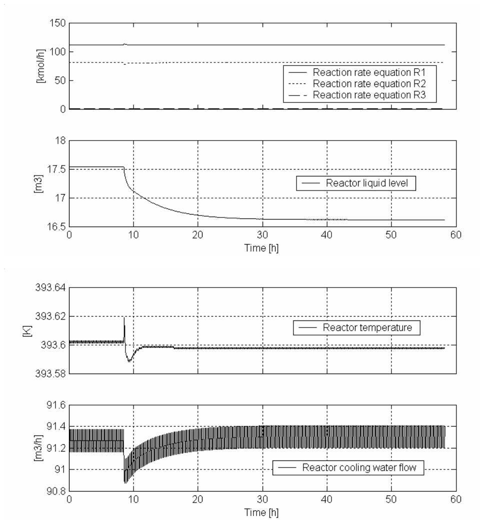
Same control profile

Lower pressure – leads to larger gas phase (reactor) volume

Less compressor load

76

TE Case Study – Results I



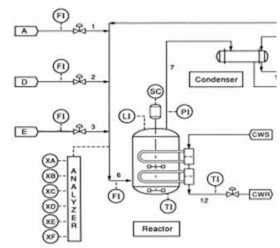
Shift in TE process

Same production rate

More volume for reaction

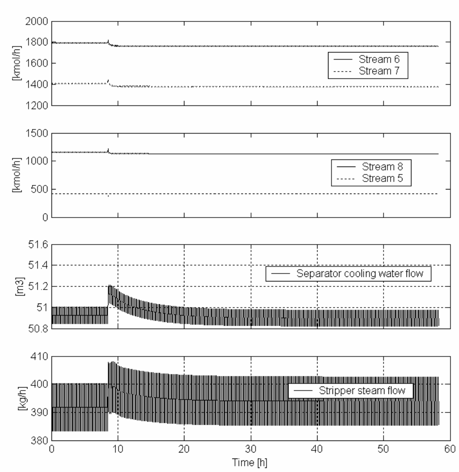
Same reactor temperature

Initially less cooling water flow (more evaporation)



77

Case Study- Results II



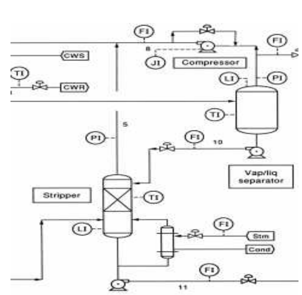
Shift in TE process

Shift in reactor effluent to more condensables

Increase cooling water flow

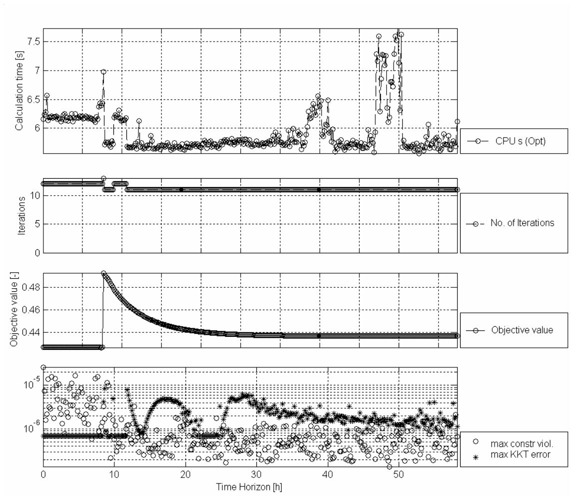
Increase stripper steam to ensure same purity

Less compressor work



78

Case Study: Change Reactor Pressure by 60 kPa



Optimization with IPOPT

1000 Optimization Cycles

5-7 CPU seconds

11-14 Iterations

Optimization with SNOPT

Often failed due to poor conditioning

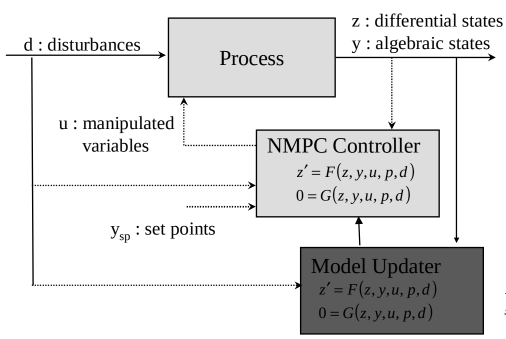
Could not be solved within sampling times

> 100 Iterations

79

Parameter/State Estimation

NMPC Estimation and Control



Moving Horizon Estimation?

- Estimate a finite number of states and model parameters (unmeasured disturbances, rate constants, transport parameters)
- Compensate for process drifts and slowly changing conditions
- Allow better controller performance

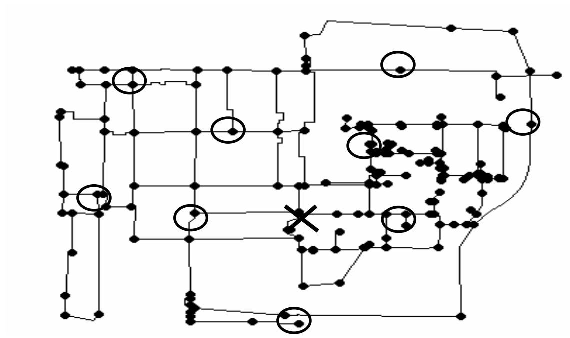
Parameter Estimation Subproblem

$$\min_{\boldsymbol{p}} \quad \sum \| y(t) - \hat{y} \|_{Q^y} + \sum \| u^k - \hat{u}^k \|_{Q^u}$$

$$\begin{aligned} s.t. \quad & z'(t) = F(z(t), y(t), \boldsymbol{p}, u(t), t) \\ & 0 = G(z(t), y(t), \boldsymbol{p}, u(t), t) \\ & z(t) = z^{\text{init}} \\ & \text{Bound Constraints} \\ & \text{Other Constraints} \end{aligned}$$

80

Early Warning Detection System Municipal Water Networks



•Installed sensors provide an early warning of contamination

•System provides only a coarse measure of contamination time and location

•Desired: Accurate and fast time & location information

81

Optimization Problem

Node Concentrations & Injection Terms Only

$$\min_{m(t),\bar{c}(x,t),\hat{c}(t)} \Psi = \sum_{r\in\Theta_s} \sum_{k\in\mathcal{N}_s} \frac{1}{2}\int_0^{t_f} w_k(t)\left(\hat{c}_k(t)-\hat{c}_k^{\star}(t)\right)^2\ \delta(t-t_r)\ dt + \frac{\rho}{2}\int_0^{t_f} m_k(t)^2 dt$$

$$\left.\begin{aligned} \frac{\partial\bar{c}_i(x,t)}{\partial t} + u_i(t)\frac{\partial\bar{c}_i(x,t)}{\partial x} &= 0,\\ \bar{c}_i(x{=}\mathcal{I}_i(t),t) &= \hat{c}_{k_i(t)}(t),\\ \bar{c}_i(x,t{=}0) &= 0, \end{aligned}\right\} \forall i\in\mathcal{P},$$

Only Constraints with Spatial Dependence

$$\hat{c}_k(t) = \frac{\left(\sum_{i\in\Gamma_k(t)} Q_i(t)\ \bar{c}_i(x{=}\mathcal{O}_i(t),t)\right) + m_k(t)}{\left(\sum_{i\in\Gamma_k(t)} Q_i(t)\right) + Q_k^{ext}(t) + Q_k^{inj}(t)}, \qquad \forall k\in\mathcal{J},$$

Pipe Boundary Concentrations

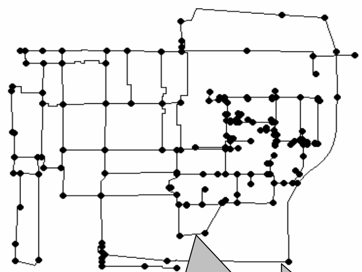
$$\left.\begin{aligned} V_k(t)\frac{d\hat{c}_k(t)}{dt} &= \left(\sum_{i\in\Gamma_k(t)} Q_i(t)\ \bar{c}_i(x{=}\mathcal{O}_i(t),t)\right) + m_k(t) - \left[\left(\sum_{i\in\Gamma_k(t)} Q_i(t)\right) + Q_k^{ext}(t) + Q_k^{inj}(t)\right]\hat{c}_k(t),\\ \hat{c}_k(t{=}0) &= 0, \end{aligned}\right\} \forall k\in\mathcal{S},$$

$$m_k(t) \geq 0, \qquad \forall k\in\mathcal{N}.$$

Injection Terms Only

82

Water Quality Model



Pipes, Valves, Pumps

Collapsed Node Models

Plug Flow

Complete Mixing

No Reaction

Known Sources Contaminant Free

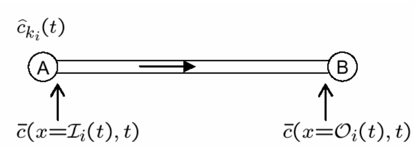
Time Dependent Mass Injections at All Nodes (Negligible Flow rates)

Storage Tanks, Junctions

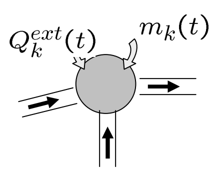
Decoupled Hydraulics and Water Quality Calculations

83

Modeling Water Distribution Systems



$$\frac{\partial \bar{c}_i(x,t)}{\partial t} + u_i(t)\frac{\partial \bar{c}_i(x,t)}{\partial x} = 0,$$
$$\bar{c}_i(x{=}\mathcal{I}_i(t),t) = \hat{c}_{k_i(t)}(t),$$
$$\bar{c}_i(x,t{=}0) = 0,$$



$$\hat{c}_k(t) = \frac{\left(\sum_{i\in\Gamma_k(t)} Q_i(t)\ \bar{c}_i(x{=}\mathcal{O}_i(t),t)\right) + m_k(t)}{\left(\sum_{i\in\Gamma_k(t)} Q_i(t)\right) + Q_k^{ext}(t) + Q_k^{inj}(t)},$$

$$V_k(t)\frac{d\hat{c}_k(t)}{dt} = \left(\sum_{i\in\Gamma_k(t)} Q_i(t)\ \bar{c}_i(x{=}\mathcal{O}_i(t),t)\right) + m_k(t) - \left[\left(\sum_{i\in\Gamma_k(t)} Q_i(t)\right) + Q_k^{ext}(t) + Q_k^{inj}(t)\right]\hat{c}_k(t),$$

$$\hat{c}_k(t{=}0) = 0,$$

84

Existing Simulation Techniques

Eulerian

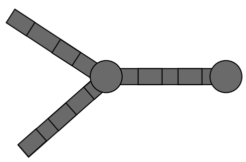
Discretize in time and space

Track concentration at fixed points or volumes

Local process for simulation, but global treatment needed for simultaneous optimization

Same as our Discretization

Too Many Constraints



Lagrangian

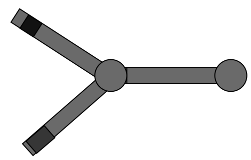
Discretize in time alone

Track concentration of elements as they move

Algorithmic in nature

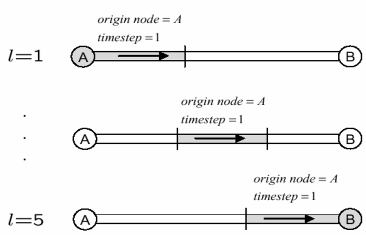
No Straightforward Representation

Derivative Calculations?



Review of these methods by Rossman and Boulos, 1996.

Origin Tracking Algorithm



$l=1$

$$\bar{c}(x=\mathcal{I}_i(t_1), t_1) = \hat{c}_A(t_1)$$
$$\bar{c}(x=\mathcal{O}_i(t_1), t_1) = 0$$

$l=5$

$$\bar{c}(x=\mathcal{I}_i(t_5), t_5) = \hat{c}_A(t_5)$$
$$\bar{c}(x=\mathcal{O}_i(t_5), t_5) = \hat{c}_A(t_1)$$

Known Hydraulics – Function of Time

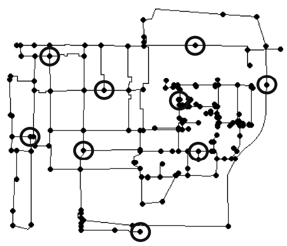
Pipe Network PDEs Linear in Concentration

Pipe by Pipe PDEs

- Efficient for Large Networks
- Convert PDEs to DAEs with variable time delays

Removes Need to Discretize in Space

Source Inversion Formulation



Water network demands known.

Linear PDEs for concentration

Convert to delay-differential equations

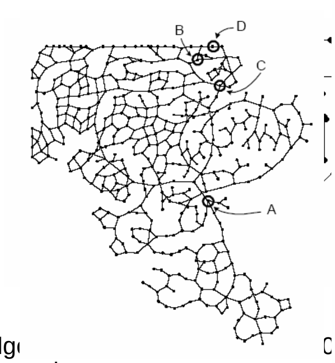
Apply implicit R-K formulation

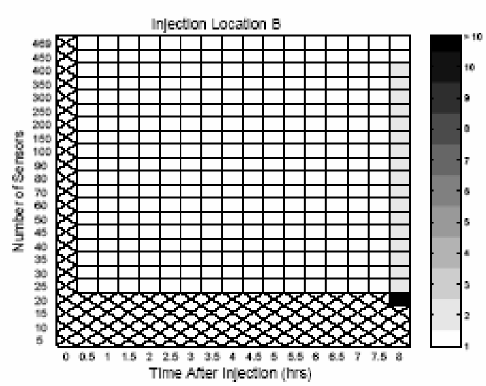
Flow demands → Hydraulic Simulator (EPANET) → Formulation Tool →

$$\begin{aligned} \min_{\bar{c},c,m} \quad & f(c,m) \\ \text{s.t.} \quad & \bar{c} - Pc = 0, \\ & \bar{N}\bar{c} + Nc + Mm = 0, \\ & m \geq 0 \end{aligned}$$

87

Municipal Source Detection Example





Alg
networks

Solution time < 2 CPU minutes for ~ 250,000 variables, ~45,000 degrees of freedom

- Effective in a real time setting

Formulation tool links to existing water network software

Can impose unique solutions through an extended MIQP formulation (post-processing phase)

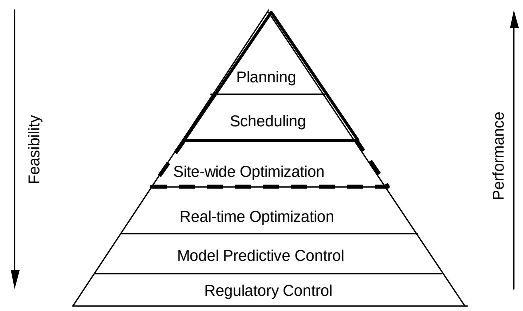
88

Decision Pyramid for Process Operations

Goal: Bridge between planning, logistics (linear, discrete problems) and detailed process models (nonlinear, spatial, dynamic)

Planning and Scheduling

- Many Discrete Decisions
- Few Nonlinearities

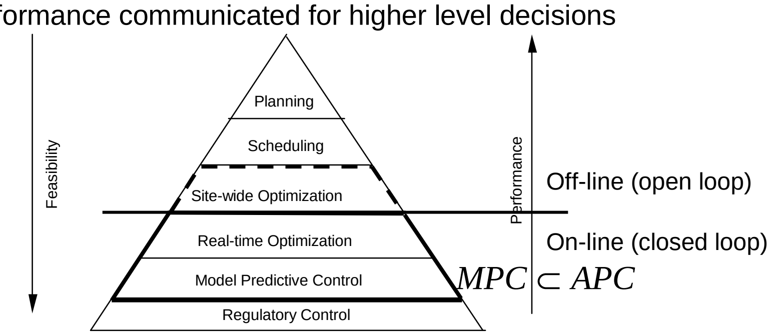


89

Decision Pyramid for Process Operations

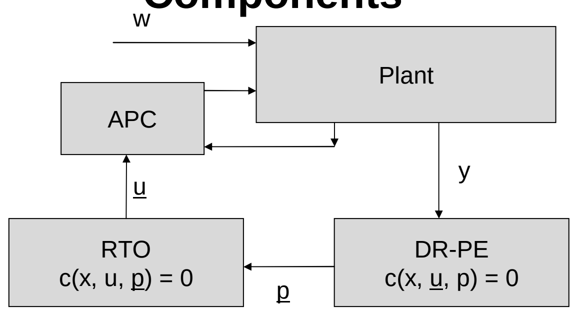
Real-time Optimization and Advanced Process Control

- Fewer discrete decisions
- Many nonlinearities
- Frequent, “on-line” time-critical solutions
- Higher level decisions must be feasible
- Performance communicated for higher level decisions



90

Steady-state On-line Optimization: Components



- Data reconciliation – identify gross errors and inconsistency in data
- Periodic update of process model identification
- Usually requires APC loops (MPC, DMC, etc.)
- RTO/APC interactions: Assume decomposition of time scales
  - APC to handle disturbances and fast dynamics
  - RTO to handle static operations
- Typical cycle: 1-2 hours, closed loop

- What if steady state and dynamic models are inconsistent?

6

Dynamic Real-time Optimization

Integrate On-line Optimization/Control with Off-line Planning

- Consistent, first-principle models
- Consistent, long-range, multi-stage planning
- Increase in computational complexity
- Time-critical calculations

Applications

- Batch processes
- Grade transitions
- Cyclic reactors (coking, regeneration…)
- Cyclic processes (PSA, SMB…)

Continuous processes are never in steady state:

- Feed changes
- Nonstandard operations
- Optimal disturbance rejections

Simulation environments (e.g., ACM, gPROMS) and first principle dynamic models are widely used for off-line studies

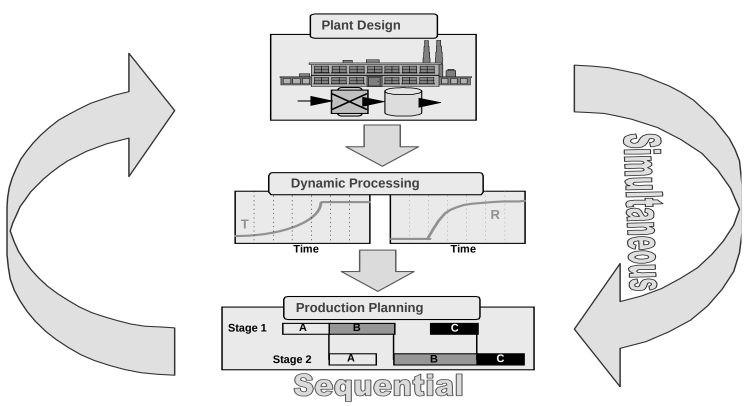
8 92

Some DRTO Case Studies

- Integrated grade transitions
  - MINLP of scheduling with dynamics (Flores & Grossmann, 2006, Prata et al., 2007)
  - *Significant reduction in transition times*
- Dynamic Predictive Scheduling
  - Processes and supply chains need to optimally respond to disturbances through dynamic models
  - *Reduction in energy cost by factor of two*
- Cyclic Process Optimization
  - Decoking scheduling
  - SMB optimization
  - PSA optimization
  - *Productivity increases by factor of 2-3*

9

Batch Integration Case Study



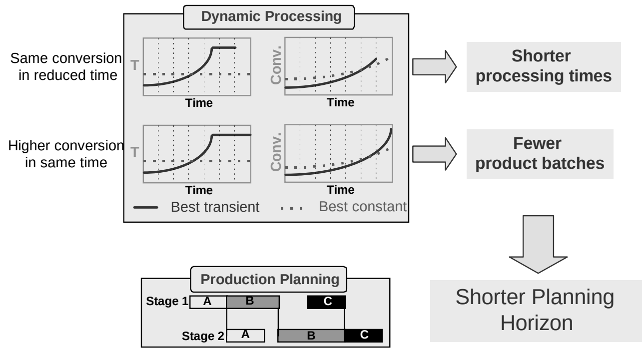
•What are the Interactions between Design and Dynamics and Planning?

•What are the differences between Sequential and Simultaneous Strategies?

•Especially Important in Batch Systems

94

Simultaneous Dynamic Optimization

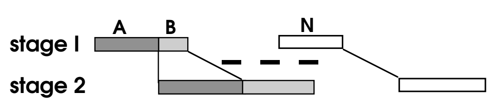


- discretize (DAEs), state and control profiles
- large-scale optimization problem
- handles profile constraints directly
- incorporates equipment variables directly
- DAE model solved only once
- converges for unstable systems

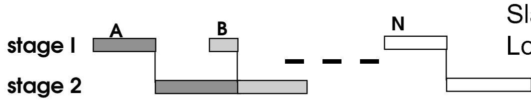
Scheduling Formulation

- sequencing of tasks, products equipment
- expensive discrete combinatorial optimization
- consider ideal transfer policies (UIS and ZW)
- closed form relations (Birewar and Grossmann, 1989)

Unlimited Int. Storage(UIS)
Short production cycle
Cycle time independent of sequence



Zero Wait (ZW)
Immediate transfer required
Slack times dependent on pair
Longer production cycle required



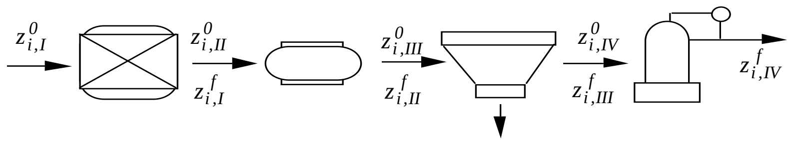
Case Study Example

4 stages, 3 products of different purity

Dynamic reactor - temperature profile

Dynamic column - reflux profile

$$A + B \rightarrow C$$
$$C + B \rightarrow P + E$$
$$P + C \rightarrow G$$



Process Optimization Cases

SQ - Sequential Design - Scheduling - Dynamics

SM - Simultaneous Design and Scheduling

Dynamics with endpoints fixed .

SM* - Simultaneous Design, Scheduling and Dynamics

97

Scenarios in Case Study

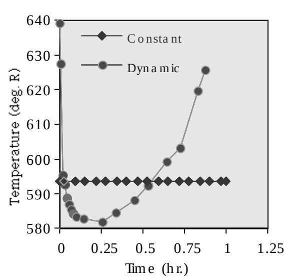
Comparison of Dynamic vs. Best Constant Profiles

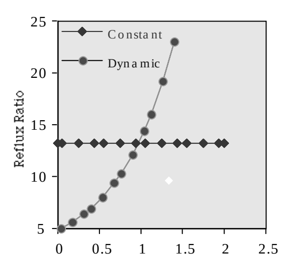
R0 - best constant temperature profile

R1 - optimal temperature policy

C0 - best constant reflux ratio

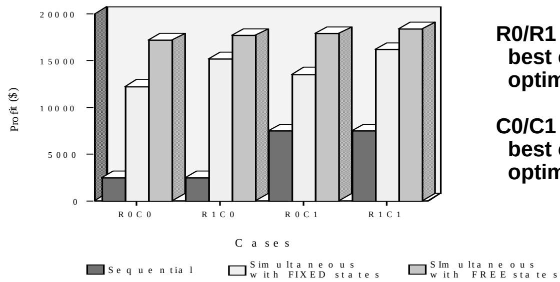
C1 - optimal reflux ratio





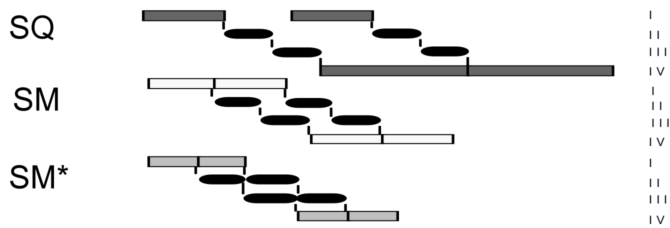
98

Results for Simultaneous Cases



R0/R1
best constant /
optimal temperature

C0/C1
best constant /
optimal reflux ratio



- ZW schedule becomes tighter

- less dependent on product sequences

99

Summary

<u>Sequential</u> <u>Approaches</u>

- Parameter Optimization
  - Gradients by: Direct and Adjoint Sensitivity Equations
- Optimal Control (Profile Optimization)
  - Variational Methods
  - NLP-Based Methods
- Require Repeated Solution of Model
- State Constraints are Difficult to Handle

<u>Simultaneous</u> <u>Approach</u>

- Discretize ODE's using orthogonal collocation on finite elements (solve larger optimization problem)
- Accurate approximation of states, location of control discontinuities through element placement.
- Straightforward addition of state constraints.
- Deals with unstable systems

<u>Simultaneous Strategies are Effective</u>

- Directly enforce constraints
- Solve model only once
- Avoid difficulties at intermediate points

<u>Large-Scale Extensions</u>

- Exploit structure of DAE discretization through decomposition
- Large problems solved efficiently with IPOPT

100

DAE Optimization Resources

References

Betts, J. T., Practical Methods for Optimal Control Using Nonlinear Programming, SIAM, (2001)

Bryson, A.E. and Y.C. Ho, Applied Optimal Control, Ginn/Blaisdell, (1968).

Himmelblau, D.M., T.F. Edgar and L. Lasdon, Optimization of Chemical Processes, McGraw-Hill, (2001).

Ray. W.H., Advanced Process Control, McGraw-Hill, (1981).

Software

• Dynamic Optimization Codes

ACM – Aspen Custom Modeler

DynoPC - simultaneous optimization code (CMU)

COOPT - sequential optimization code (Petzold)

gOPT - sequential code integrated into gProms (PSE)

MUSCOD - multiple shooting optimization (Bock)

NOVA - SQP and collocation code (DOT Products)

• Sensitivity Codes for DAEs

DASOLV - staggered direct method (PSE)

DASPK 3.0 - various methods (Petzold)

SDASAC - staggered direct method (sparse)

DDASAC - staggered direct method (dense)

101

References

F. Allgöwer and A. Zheng (eds.), *Nonlinear Model Predictive Control*, Birkhaeuser, Basel (2000)

R. D. Bartusiak, "NLMPC: A platform for optimal control of feed- or product-flexible manufacturing," in *Nonlinear Model Predictive Control 05,* Allgower, Findeisen, Biegler (eds.), Springer, to appear

Biegler Homepage: http://dynopt.cheme.cmu.edu/papers.htm

Forbes, J. F. and Marlin, T. E.. Model Accuracy for Economic Optimizing Controllers: The Bias Update Case. *Ind.Eng.Chem.Res*. 33, 1919-1929. 1994

Forbes, J. F. and Marlin, T. E.. "Design Cost: A Systematic Approach to Technology Selection for Model-Based Real-Time Optimization Systems," *Computers Chem.Engng.* 20[6/7], 717-734. 1996

Grossmann Homepage: http://egon.cheme.cmu.edu/papers.html

M. Grötschel, S. Krumke, J. Rambau (eds.), *Online Optimization of Large Systems,* Springer, Berlin (2001)

K. Naidoo, J. Guiver, P. Turner, M. Keenan, M. Harmse "Experiences with Nonlinear MPC in Polymer Manufacturing," in *Nonlinear Model Predictive Control 05,* Allgower, Findeisen, Biegler (eds.), Springer, to appear

Yip, W. S. and Marlin, T. E. "Multiple Data Sets for Model Updating in Real-Time Operations Optimization," *Computers Chem.Engng*. 26[10], 1345-1362. 2002.

42

References – Some DRTO Case Studies

Busch, J.; Oldenburg, J.; Santos, M.; Cruse, A.; Marquardt, W. Dynamic predictive scheduling of operational strategies for continuous processes using mixed-logic dynamic optimization, *Comput. Chem. Eng.,* **2007**, 31, 574-587.

Flores-Tlacuahuac, A.; Grossmann, I.E. Simultaneous cyclic scheduling and control of a multiproduct CSTR, *Ind. Eng. Chem. Res.*, **2006**, 27, 6698-6712.

Kadam, J.; Srinivasan, B., Bonvin, D., Marquardt, W. Optimal grade transition in industrial polymerization processes via NCO tracking. *AIChE J.*, **2007**, 53, 3, 627-639.

Oldenburg, J.; Marquardt, W.; Heinz D.; Leineweber, D. B., Mixed-logic dynamic optimization applied to batch distillation process design, *AIChE J.* **2003**, 48(11), 2900- 2917.

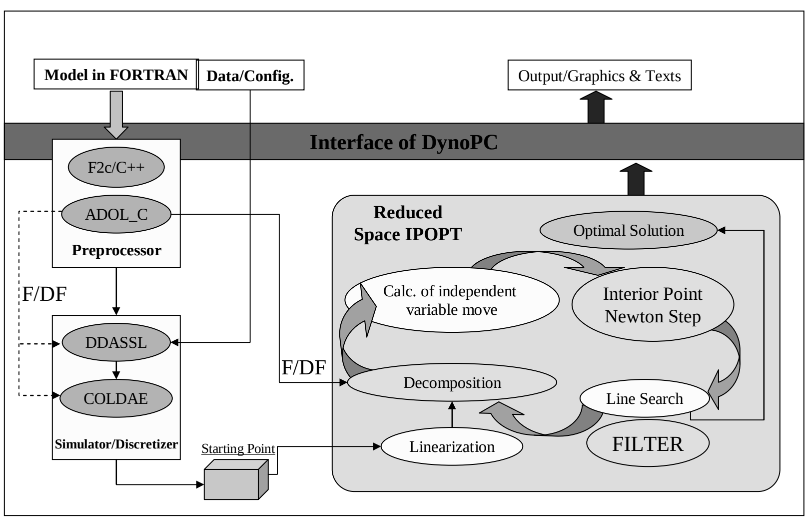
E Perea, B E Ydstie and I E Grossmann, A model predictive control strategy for supply chain optimization, *Comput. Chem. Eng.,* **2003**, 27, 1201-1218.

M. Liepelt, K Schittkowski, Optimal control of distributed systems with breakpoints, p. 271 in M. Grötschel, S. Krumke, J. Rambau (eds.), *Online Optimization of Large Systems*, Springer, Berlin (2001)

See also Biegler homepage

43

DynoPC – Windows Implementation

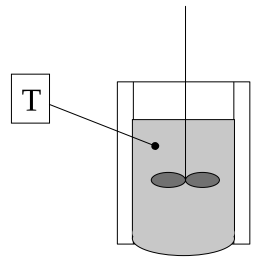


104

Example: Batch Reactor Temperature

$$A \xrightarrow{1} B \xrightarrow{2} C$$

$$Max \quad b(t_f)$$



$s.t.$

$$\frac{da}{at} = -k_1 \exp(-\frac{E_1}{RT}) \cdot a$$

$$\frac{db}{at} = k_1 \exp(-\frac{E_1}{RT}) \cdot a - k_2 \exp(-\frac{E_2}{RT}) \cdot b$$

a + b + c = 1

105

Example: Car Problem

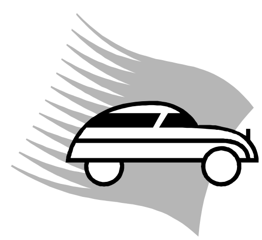
Min t_f

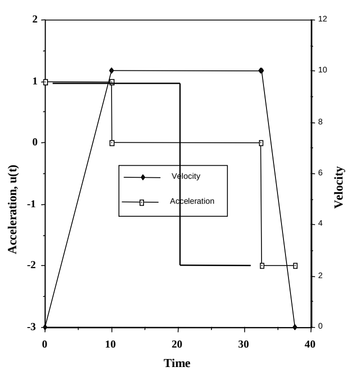
s.t. $z_1' = z_2$

$z_2' = u$

$z_2 \leq z_{max}$

$-2 \leq u \leq 1$

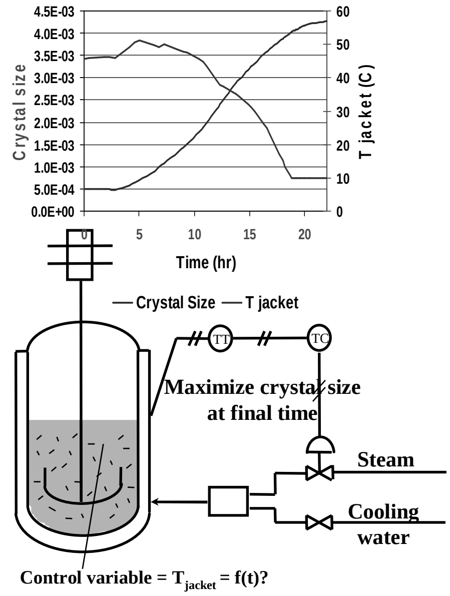





```
subroutine model(nz,ny,nu,np,t,z,dmz,y,u,p,f)
double precision t, z(nz),dmz(nz), y(ny),u(nu),p(np)

double precision f(nz+ny)

f(1) = p(1)*z(2) - dmz(1)
f(2) = p(1)*u(1) - dmz(2)

return
end
```

106

Example: Crystallizer Temperature




```
SUBROUTINE model(nz,ny,nu,np,x,z,dmz,y,u,p,f)
implicit double precision (a-h,o-z)
double precision f(nz+ny),z(nz),dmz(nz),Y(ny),yp(4),u(1)
double precision kgr, ln0, ls0, kc, ke, kex, lau, deltT, alpha
dimension a(0:3), b(0:3)
data alpha/1.d-4/,a/-66.4309d0, 2.8604d0, -.022579d0, 6.7117d-5/,
+  b/16.08852d0, -2.708263d0, .0670694d0, -3.5685d-4/, kgr/ 4.18d-3/,
+  en / 1.1d0/, ln0/ 5.d-5/, Bn / 3.85d2/, em / 5.72/, ws0/ 2.d0/,
+  Ls0/ 5.d-4     /, Kc / 35.d0 /, Kex/ 65.d0/, are/ 5.8d0 /,
+  amt/ 60.d0  /, V0 / 1500.d0/, cw0/ 80.d0/,cw1/ 45.d0/,v1 /200.d0/,
+                   tm1/ 55.d0/,x6r/0.d0/, tem/ 0.15d0/,clau/ 1580.d0/,lau/1.35d0/,
+   cp/ 0.4d0    /,cbata/ 1.2d0/, calfa/ .2d0   /, cwt/ 10.d0/

ke = kex*area
x7i = cw0*lau/(100.d0-cw0)
v = (1.d0 - cw0/100.d0)*v0
w = lau*v0
yp(1) = (deltT + dsqrt(deltT**2 + alpha**2))*0.5d0
yp(2) = (a(0) + a(1)*yp(4) + a(2)*yp(4)**2 + a(3)*yp(4)**3)
yp(3) = (b(0) + b(1)*yp(4) + b(2)*yp(4)**2 + b(3)*yp(4)**3)
deltT = yp(2) - z(8)
yp(4) = 100.d0*z(7)/(lau+z(7))

f(1) = Kgr*z(1)**0.5*yp(1)**en - dmz(1)
f(2) = Bn*yp(1)**em*1.d-6 - dmz(2)
f(3) = ((z(2)*dmz(1) + dmz(2) * Ln0)*1.d+6*1.d-4) - dmz(3)
f(4) = (2.d0*cbata*z(3)*1.d+4*dmz(1)+dmz(2)*Ln0**2*1.d+6)-dmz(4)
f(5) = (3.d0*calfa*z(4)*dmz(1)+dmz(2)*Ln0**3*1.d+6) - dmz(5)
f(6) = (3.d0*Ws0/(Ls0**3)*z(1)**2*dmz(1)+clau*V*dmz(5))-dmz(6)
f(7) = -dmz(6)/V - dmz(7)
f(8) = (Kc*dmz(6) - Ke*(z(8) - u(1)))/(w*cp) - dmz(8)
f(9) = y(1)+YP(3)- u(1)
return
end
```

107