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

$$
\begin{aligned}
\min \quad & \Phi(z(t_f)) \\
s.t.\ dz(t)/dt &= f(z(t), y(t), u(t), t, p), \\
z(0) &= z_0 \\
0 &= g(z(t), y(t), u(t), t, p) \\
& z_l \le z(t) \le z_u \\
& y_l \le y(t) \le y_u \\
& u_l \le u(t) \le u_u \\
& p_l \le p \le p_u
\end{aligned}
$$

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

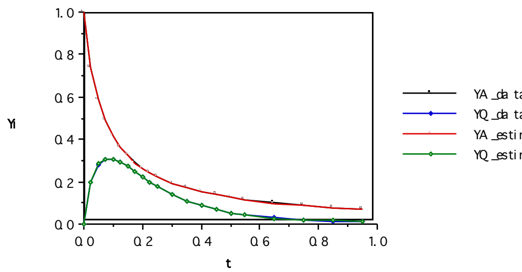
Catalytic Cracking of Gasoil (Tjoa, 1991)

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

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

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

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



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

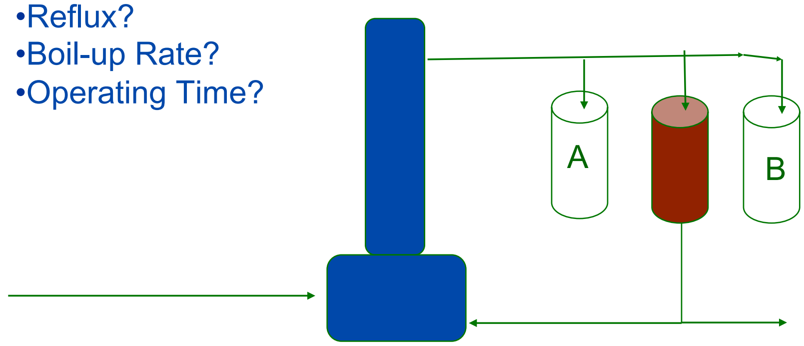
Objective Function: Ordinary Least Squares

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

5

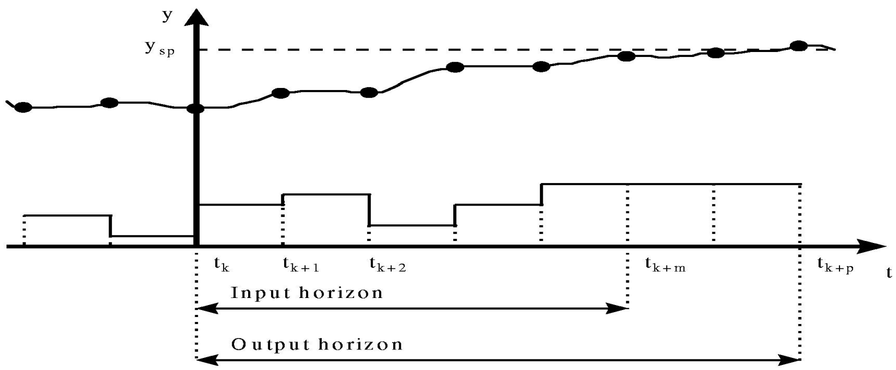
Batch Distillation Multi-product Operating Policies

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



6

Nonlinear Model Predictive Control (NMPC)



$$\min_{u} \sum_{j} \| y(t_{k+j}) - y^{sp} \|^2_{Q_y} + \sum_{j} \| \boldsymbol{u}(\boldsymbol{t}_{k+j}) - \boldsymbol{u}(\boldsymbol{t}_{k+j-1}) \|^2_{Q_u}$$

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

Bound Constraints

Other Constraints

Batch Process Optimization

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

DAE models:

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

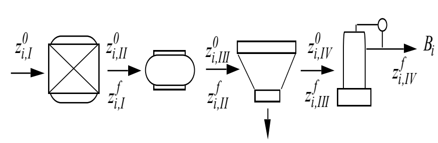
$g(z, y, u, p) = 0$

$$\boxed{\begin{aligned} & A + B \rightarrow C \\ & C + B \rightarrow P + E \\ & P + C \rightarrow G \end{aligned}}$$

Number of states and DAEs: $n_z + n_y$

Parameters for equipment design (reactor, column)

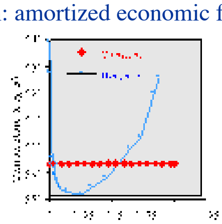
n_u control profiles for optimal operation



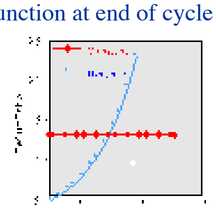
Constraints: $u_L \le u(t) \le u_U$ $\quad z_L \le z(t) \le z_U$

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

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



optimal reactor temperature policy



optimal column reflux ratio

Reactor Optimization Example

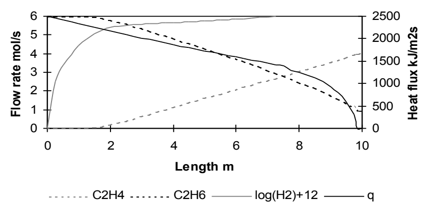
The cracking furnace is an important example in the olefin production industry, where various hydrocarbon feedstocks react. Consider a simplified model for ethane cracking (Chen et al., 1996). The objective is to find an optimal profile for the heat flux along the reactor in order to maximize the production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

$$\begin{aligned} &\text{Max } F(C_2H_4)_{exit} \\ &\text{s.t. DAE Model} \\ &\quad T_{exit} \leq 1180 \text{ K} \end{aligned}$$

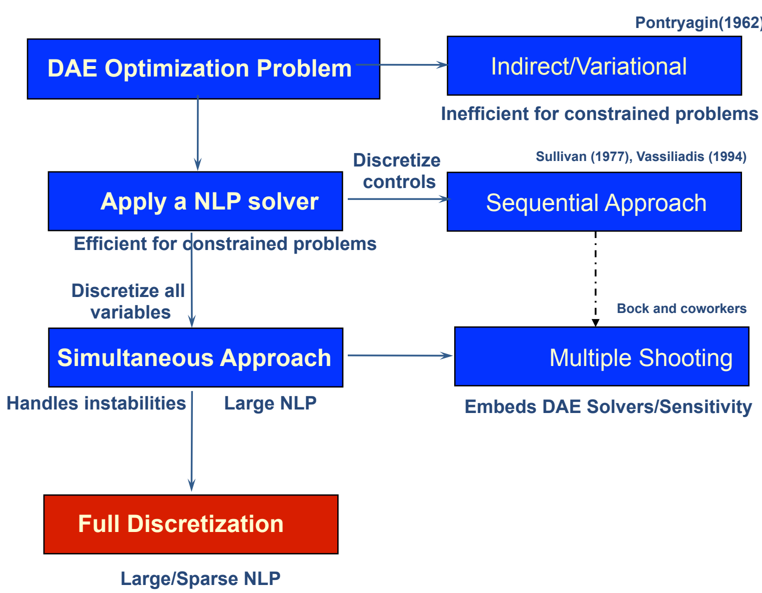
The reaction system includes six molecules, three free radicals, and seven reactions. The model also includes the heat balance and the pressure drop equation. This gives a total of eleven differential equations.

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

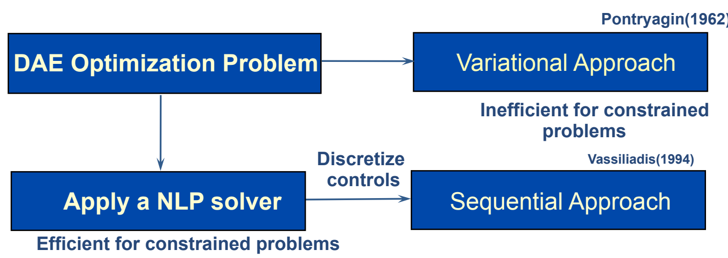
Concentration and Heat Addition Profile



Dynamic Optimization Approaches



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

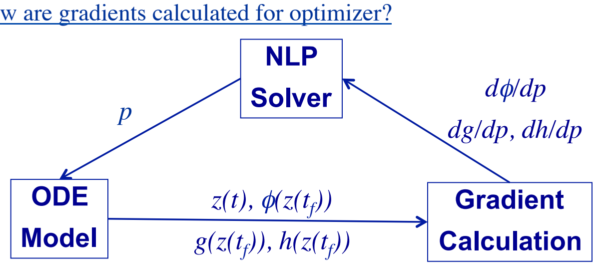
$$Min \quad \Phi\,(z(t_f))$$

$$dz/dt = f(z, p),\ z\,(0) = z_0$$

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

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?



Gradient Calculation

Perturbation

Sensitivity Equations

Adjoint Equations

Perturbation

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

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

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

- 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)

13

Direct Sensitivity

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

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

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

(nz x np sensitivity equations)

• z and s_i , i = 1,…np, can 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 parameter variables (1 + ng + nh > np)

14

Example: Sensitivity Equations

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

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

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

$$s(t)_{a,j} = \partial z(t)_j / \partial p_a, \quad s(t)_{b,j} = \partial z(t)_j / \partial p_b, j = 1,2$$

$$s_{a,1}' = 2z_1 s_{a,1} + 2z_2 s_{a,2}$$

$$s_{a,2}' = z_1 s_{a,2} + z_2 s_{a,1} + s_{a,1} p_b$$

$$s_{a,1}(0) = 0, s_{a,2}(0) = 1$$

$$s_{b,1}' = 2z_1 s_{b,1} + 2z_2 s_{b,2}$$

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

$$s_{b,1}(0) = 0, s_{b,2}(0) = 0$$

Adjoint Sensitivity

Adjoint or Dual approach to sensitivity

Adjoin model to objective function or constraint

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

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

($\lambda(t)$) 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 not 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 forward

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

17

Example: Adjoint Equations

$$z_1' = z_1^2 + z_2^2$$
$$z_2' = z_1 z_2 + z_1 p_b$$
$$z_1(0) = 5, z_2(0) = p_a$$

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

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

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

then becomes:

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

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

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

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

18

Example: Hot Spot Reactor

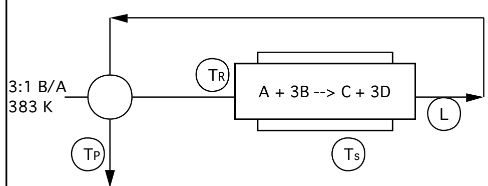
$$\underset{T_P, T_R, L, T_S}{Min} \quad \Phi = L - \int_0^L (T(t) - T_S / T_R)dt$$

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

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

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

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



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) ≤ 1.45

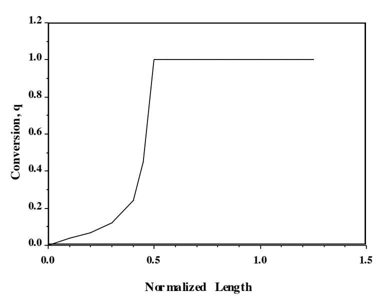
19

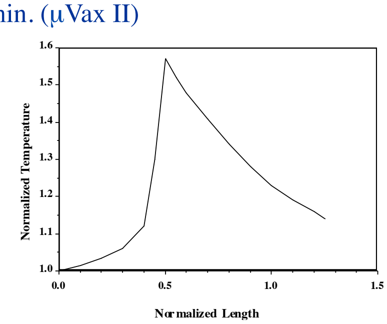
Hot Spot Reactor: Unconstrained Case

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 ≤ 1.45): no solution for sequential method (without constraint reformulation)

20

Tricks to generalize classes of problems

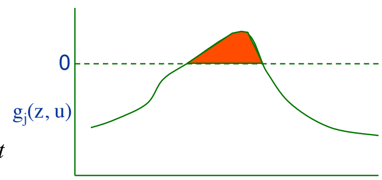
Variable Final Time (Miele, 1980)

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

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

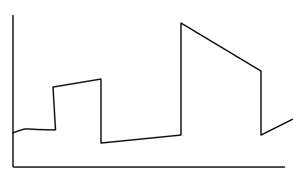
21

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 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

22

Derivation of Variational Conditions
Indirect Approach

Optimality Conditions (Bound constraints on u(t))

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

Form Lagrange function - adjoin objective function and constraints:

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

Integrate by parts:

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

23

Derivation of Variational Conditions

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

$$d\phi = \left[\frac{\partial \phi}{\partial z} + \frac{\partial g}{\partial z}\mu + \frac{\partial h}{\partial z}\nu - \lambda\right]^T \delta z(t_f) + \lambda^T(0)\delta z(0)$$
$$+ \int_0^{t_f} \left[\dot{\lambda} + \frac{\partial f(z,u)}{\partial z}\lambda\right]^T \delta z(t) + \left[\frac{\partial f(z,u)}{\partial u}\lambda + \alpha_b - \alpha_a\right]^T \delta u(t)\ dt \ge 0$$

0

0

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

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

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

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

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

24

Derivation of Variational Conditions

$$\dot{z} = f(z,\ u);\ z(0) = z_0$$

$$\dot{\lambda} = -\frac{\partial \mathrm{H}}{\partial z} = -\frac{\partial f}{\partial z}\lambda \quad \text{where } \mathrm{H}(z,u,\lambda) = f(z,u,\lambda)^T \lambda$$

$$\lambda(t_f) = \left\{\frac{\partial \phi}{\partial z} + \frac{\partial g}{\partial z}\mu + \frac{\partial h}{\partial z}\gamma\right\}_{t_f} \quad (\lambda(0) = 0, \text{ if } z(0) \text{ is not specified})$$

$$\frac{\partial \mathrm{H}}{\partial u} = \frac{\partial f}{\partial u}\lambda = \alpha_a - \alpha_b$$

$$0 \le \alpha_a \perp (u(t) - u_a) \ge 0$$

$$0 \le \alpha_b \perp (u_b - u(t)) \ge 0$$

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

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

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

Car Problem

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

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

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

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

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

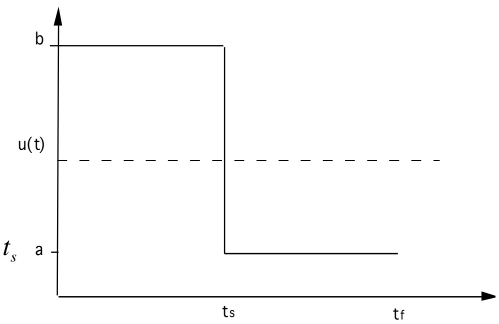
$$\dot{\lambda}_2 = -\lambda_1 ==> \lambda_2(t) = c_2 + c_1(t_f - t)$$

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

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

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

Car Problem - Variational Solution



Optimal Profile

From state equations:

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

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

•Problem is linear in u(t) and x(t) – has "bang-bang" character.

•For nonlinear and larger problems, the Euler-Lagrange (variational) conditions are solved as boundary value problems.

Apply boundary conditions at $t = t_f$:

$$x_1(t) = 1/2\left(bt_f^2 + (a-b)\left(t_f - t_s\right)^2\right) = L$$

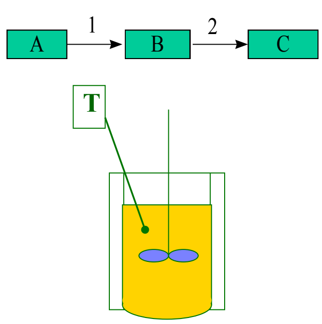
$$x_2(t) = \left(bt_s + a\left(t_f - t_s\right)\right) = 0$$

and solving for two unknowns leads to:

$$t_s = (2L/(b - b^2/a))^{1/2},\quad t_f = (1 - a/b)t_s$$

27

Batch Reactor



$$\min \quad -b(t_f)$$

$$s.t.\quad \frac{da}{dt} = -au$$

$$\frac{db}{dt} = au - kbu^{\beta}$$

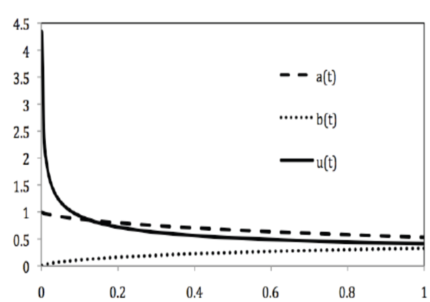
$$a(0) = 1,\quad b(0) = 0$$

$$\frac{d\lambda_1}{dt} = -(\lambda_2 - \lambda_1)u$$

$$\frac{d\lambda_2}{dt} = \lambda_2 k u^{\beta}$$

$$\lambda_1(t_f) = 0,\quad \lambda_2(t_f) = -1$$

$$\frac{\partial H}{\partial u} = J = (\lambda_2 - \lambda_1)a - \beta k \lambda_2 u^{\beta - 1} b = 0$$

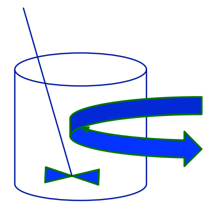


28

Batch reactor – Optimal temperature profile

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

$$
\begin{aligned}
&\quad Min \quad -b(1) \\
s.t. \quad & a' = -(u + u^2/2)a,\ a(0) = 1 \\
& b' = ua,\ b(0) = 0 \\
& 0 \le u(t) \le 5
\end{aligned}
$$



$$A \xRightarrow{u(t)} B,\ A \xRightarrow{u(t)^2/2} C$$

Optimality conditions:

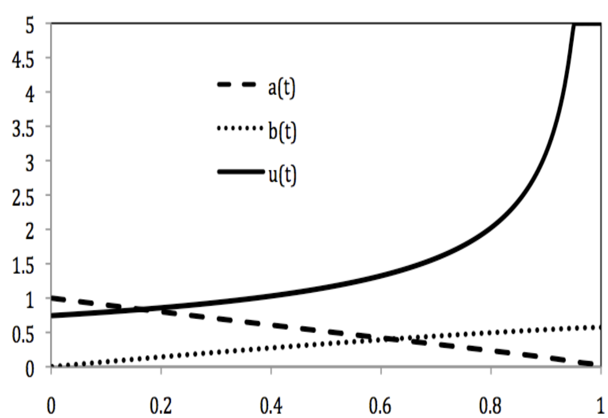
$$
\begin{gathered}
H = -\lambda_a (u + u^2/2)a + \lambda_b ua \\
\partial H / \partial u = -\lambda_a (1+u)a + \lambda_b a = \alpha_0 - \alpha_5 \\
0 \le \alpha_0 \perp u \ge 0, \quad 0 \le \alpha_5 \perp (5-u) \ge 0 \\
\lambda_a' = \lambda_a (u + u^2/2) - \lambda_b u, \quad \lambda_a(1) = 0 \\
\lambda_b'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

Cases Considered

1. NLP Approach - piecewise constant and linear profiles.
2. Indirect Approach – solve conditions as boundary value problem (BVP)

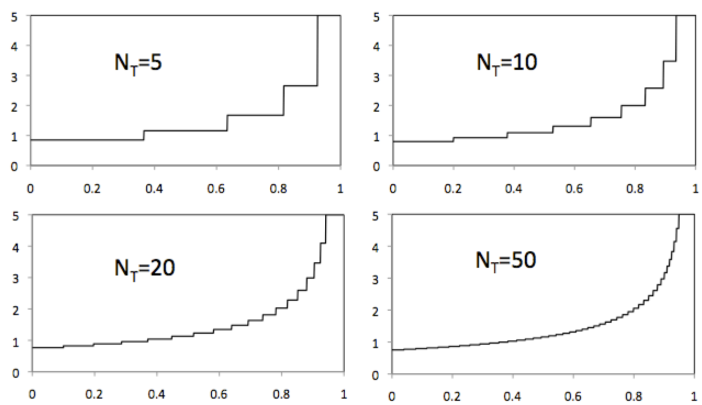
Batch Reactor Optimal Temperature Program

Indirect Approach (analytical solution)



Results: Optimum (B/A): 0.57354
Solved directly from Euler-Lagrange Equations

Batch Reactor Optimal Temperature Program
Piecewise Constant (SNOPT)



NT	Obj.	Iters.	Fcn Evals
5	0.57177	25	52
10	0.57305	47	93
20	0.57343	81	163
50	0.57352	131	227

31

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

32

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: $y_1(0) = 0$, $y_2(0) = \pi$

BVP: $y_1(0) = 0$, $y_1(1) = 0$

33

IVP Solution

If we now add round-off 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.

34

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 $\tau (e_1 + e_2)$; as a result a solution to the BVP is readily obtained.

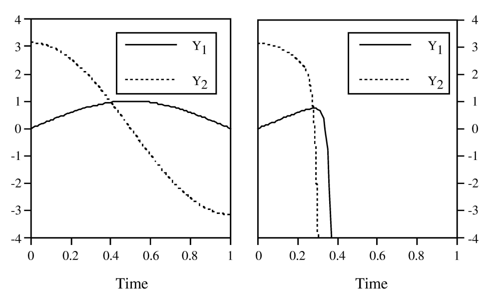
35

BVP and IVP Profiles

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

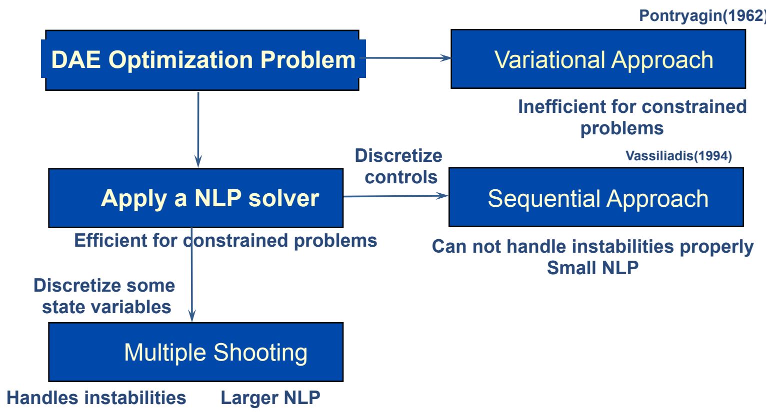
Linear BVP solves easily

IVP blows up before midpoint



36

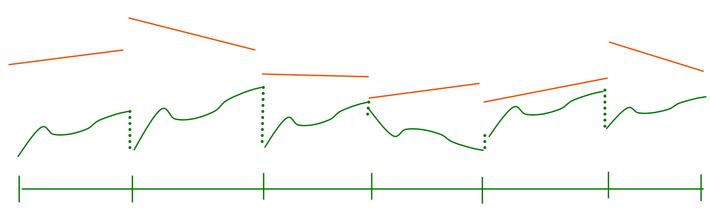
Dynamic Optimization Approaches



37

Multiple Shooting for Dynamic Optimization

Divide time domain into separate regions



Integrate DAEs state equations over each region j

Evaluate sensitivities in each region j 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 due to control profiles (u_{ij}, p) and initial conditions (z_j) in each region

38

Multiple Shooting
Nonlinear Programming Problem

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

s.t.

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

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

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

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

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

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

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

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

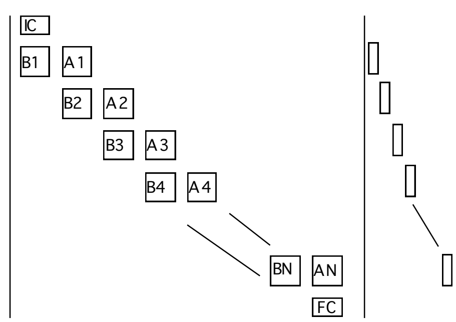
$$\min_{x \in \Re^n} f(x)$$

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

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

Solved Implicitly

BVP Problem Decomposition



Consider: Jacobian of Constraint Matrix for NLP ($A_j = -I$, $B_j = dz_{j+1}/dz_j$)

- bound unstable modes with boundary conditions (dichotomy)
- done implicitly by determining stable pivot sequences in multiple shooting
- 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?)

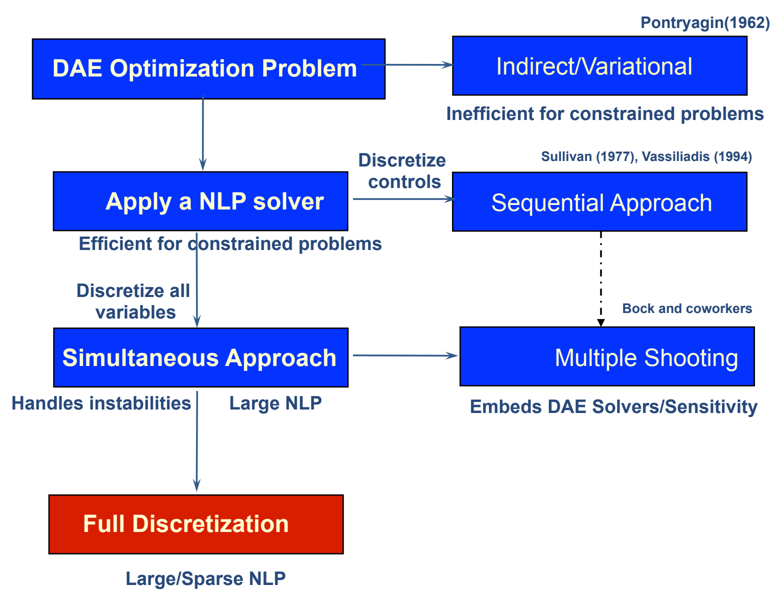
Path constraints are difficult to handle exactly for NLP approach

Block elements for each element ($B_j = dz_{j+1}/dz_j$) are dense!

Extensive developments and applications by Bock and coworkers using MUSCOD code

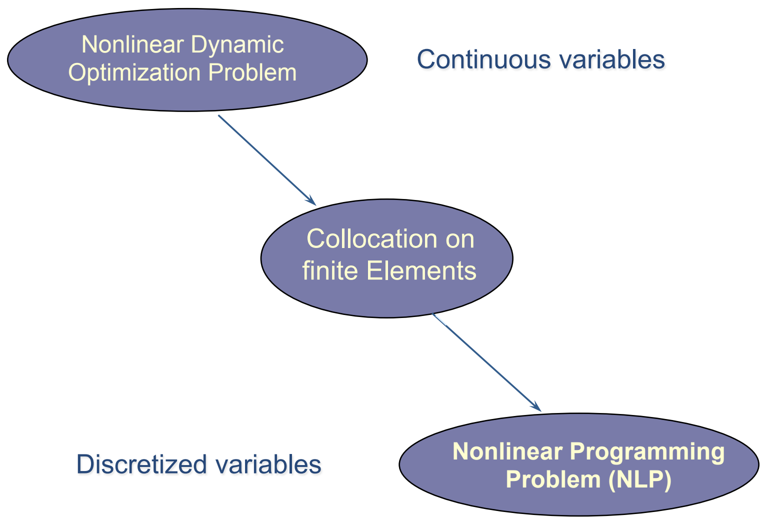
41

Dynamic Optimization Approaches



42

Nonlinear Programming Formulation



43

Discretization of Differential Equations Orthogonal Collocation

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

Approximate 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_{K+1} and u_K into ODE and apply equations at t_k.

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

44

Collocation Points Using Legendre and Radau Roots

Degree K	Legendre Roots	Radau Roots
1	0.500000	1.000000
2	0.211325	0.333333
	0.788675	1.000000
3	0.112702	0.155051
	0.500000	0.644949
	0.887298	1.000000
4	0.069432	0.088588
	0.330009	0.409467
	0.669991	0.787659
	0.930568	1.000000
5	0.046910	0.057104
	0.230765	0.276843
	0.500000	0.583590
	0.769235	0.860240
	0.953090	1.000000

Table 10.1. *Shifted Gauss-Legendre and Radau roots as collocation points.*

45

Collocation Example 1

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

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

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

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

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

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

$==> z_0 = 0$

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

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

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

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

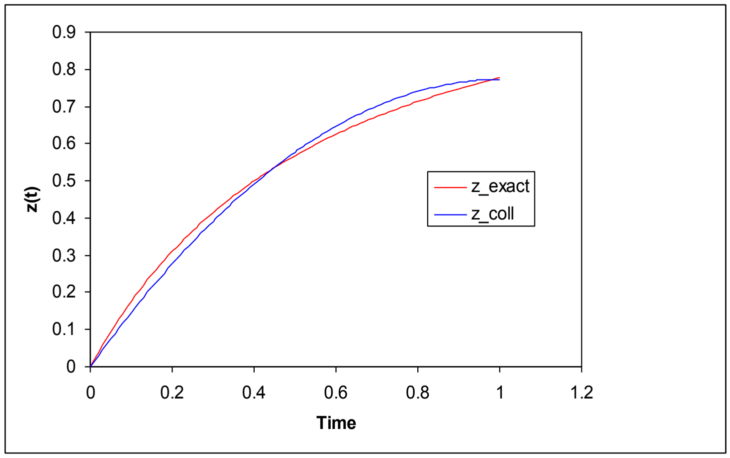
$z_0 = 0,\ z_1 = 0.291\ (0.319),\ z_2 = 0.7384\ (0.706)$

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

46

Collocation - Example 1

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



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

Converted Optimal Control Problem Using Collocation

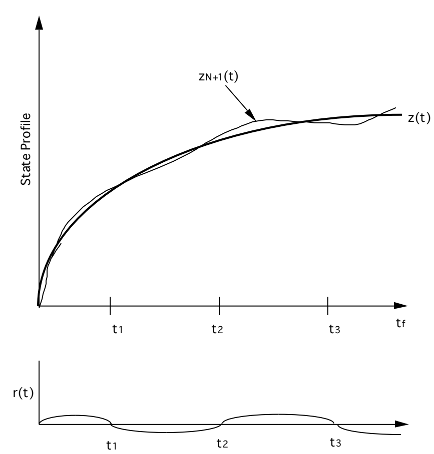
Min $\phi(z(t_f))$
s.t. $z' = f(z, u, p),\ z(0)=z_0$
$g(z(t), u(t), p) \leq 0$
$h(z(t), u(t), p) = 0$

Discretize as Nonlinear Program

Min $\phi(z_f)$

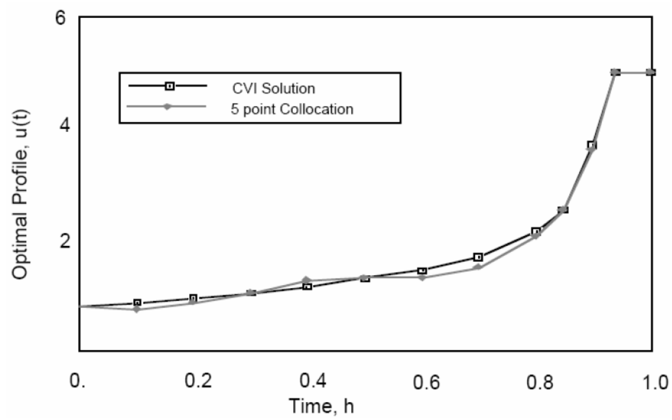
$$\left.\begin{array}{l} \sum_{j=0}^{K} z_j \dot{\ell}_j(t_k) - f(z_k, u_k) = 0,\ z_0 = z(0) \\ g(z_k, u_k) \leq 0 \\ h(z_k, u_k) = 0 \end{array}\right\} k = 1, \ldots K$$

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



How accurate is approximation

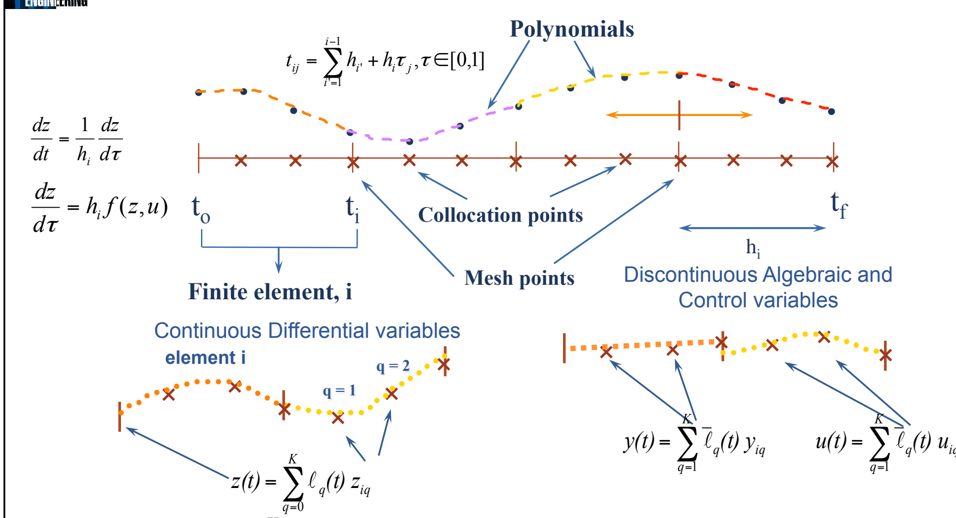
Results of Optimal Temperature Program Batch Reactor (Revisited)



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

49

Collocation on Finite Elements



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

50

Simple Model - Example 2

Example 10.2 (Demonstration of Orthogonal Collocation)

Consider a single differential equation:

$$\frac{dz}{dt} = z^2 - 2z + 1,\ z(0) = -3. \tag{10.16}$$

with $t \in [0, 1]$. This equation has an analytic solution given by $z(t) = (4t-3)/(4t+1)$. Using Lagrange interpolation and applying the collocation and continuity equations (10.7) and (10.14), respectively, with $K = 3$ collocation points, N elements and $h = 1/N$ leads to:

$$\sum_{j=0}^{3} z_{ij} \frac{d\ell_j(\tau_k)}{d\tau} = h(z_{ik}^2 - 2z_{ik} + 1),\ k = 1, \ldots, 3,\ i = 1, \ldots, N$$

51

$$z_{i+1,0} = \sum_{j=0}^{3} \ell_j(1) z_{ij}, \quad i = 1, \ldots, N-1$$

$$z_f = \sum_{j=0}^{K} \ell_j(1) z_{Nj},\ z_{1,0} = -3.$$

Using Radau collocation, we have $\tau_0 = 0$, $\tau_1 = 0.155051$, $\tau_2 = 0.644949$ and $\tau_3 = 1$. For $N = 1$, and $z_0 = -3$ the collocation equations are given by:

$$\sum_{j=0}^{3} z_j \frac{d\ell_j(\tau_k)}{d\tau} = (z_k^2 - 2z_k + 1),\ k = 1, \ldots, 3,$$

which can be written out as:

$$\begin{aligned} & z_0(-30\tau_k^2 + 36\tau_k - 9) + z_1(46.7423\tau_k^2 - 51.2592\tau_k + 10.0488) \\ & + z_2(-26.7423\tau_k^2 + 20.5925\tau_k - 1.38214) + z_3(10\tau_k^2 - \frac{16}{3}\tau_k + \frac{1}{3}) \\ & = (z_k^2 - 2z_k + 1),\ k = 1, \ldots, 3. \end{aligned}$$

Solving these three equations gives $z_1 = -1.65701$, $z_2 = 0.032053$, $z_3 = 0.207272$

52

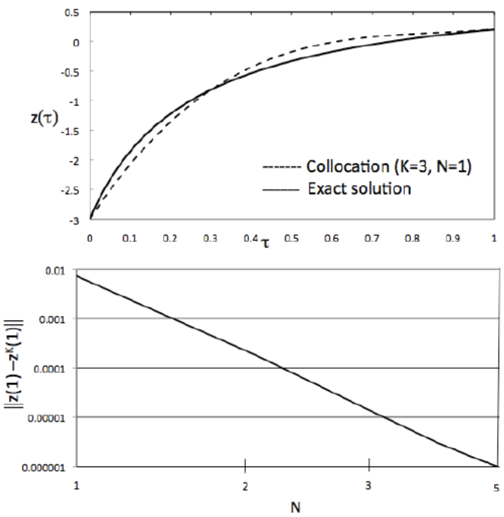


Figure 10.2. *Comparison of Radau collocation solution ($K = 3$) with exact solution for* $\frac{dz}{dt} = z^2 - 2z + 1,\ z(0) = -3$.

Nonlinear Programming Problem

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

s.t.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Hot Spot Reactor Revisited

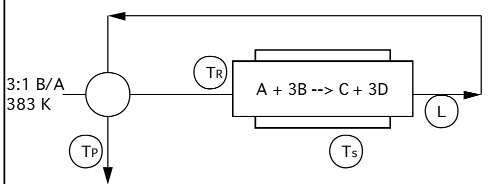
$$\underset{T_P, T_R, L, T_S}{Min} \quad \Phi = L - \int_0^L (T(t) - T_S / T_R)\, dt$$

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

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

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

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



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:

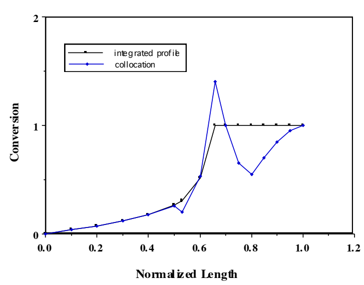
- Hot Spot - no state variable constraints
- Hot Spot with $T(t) \leq 1.45$

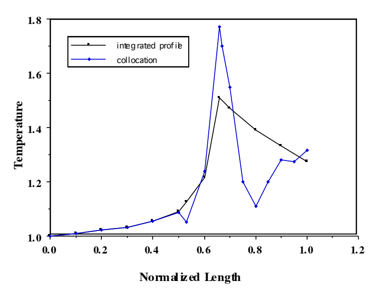
55

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





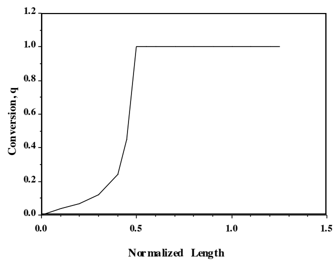
56

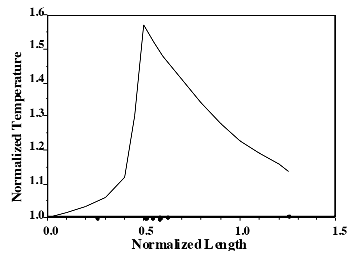
Unconstrained Case

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)
$\phi^* = -171.5$





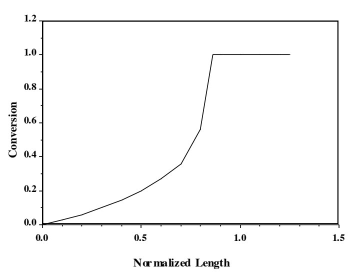
Temperature Constrained Case

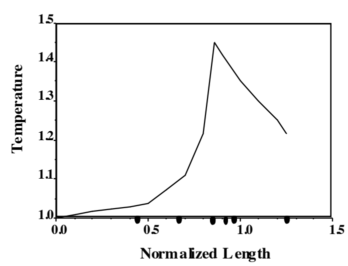
T(t) ≤ 1.45

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. Stability and Accuracy of Orthogonal Collocation

- 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. Analysis of the Optimality Conditions

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

59

Dynamic Optimization Engines

Evolution of NLP Solvers:

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

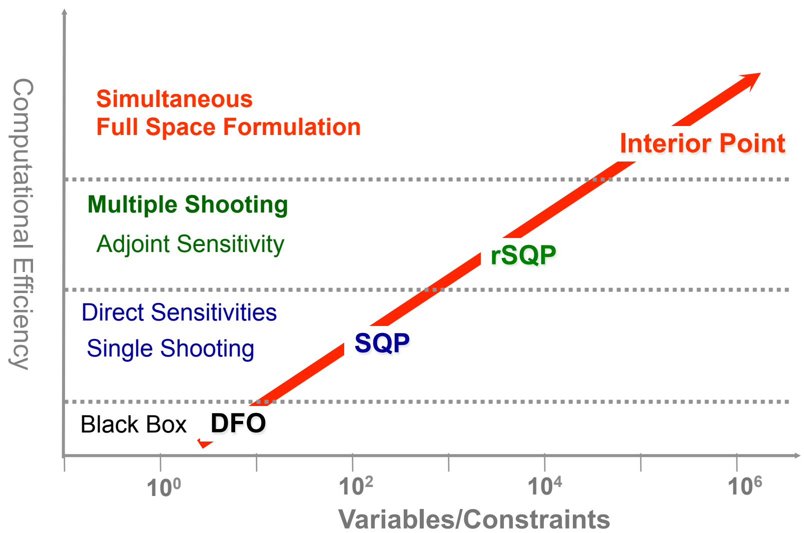
SQP ⟶ rSQP ⟶ Full-space Barrier

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.x)

60

Hierarchy of Nonlinear Programming for Dynamic Optimization Formulations



61

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	---	---	$((n_u + n_w)N)$
	$O((n_u N)^\alpha + N^2 n_w n_u + N^3 n_w n_u^2)$	$O((n_u N)^\alpha + N n_w^3 + N n_w (n_w + n_u)^2)$	$O((n_u + n_w)N)^\beta$

62

Simultaneous DAE Optimization

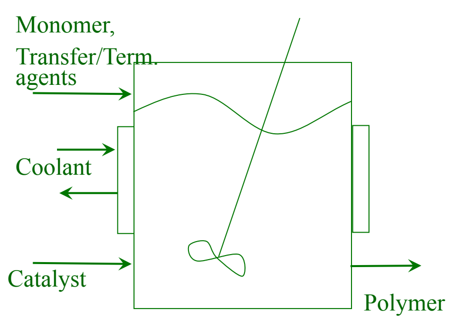
Case Studies

- 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

63

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

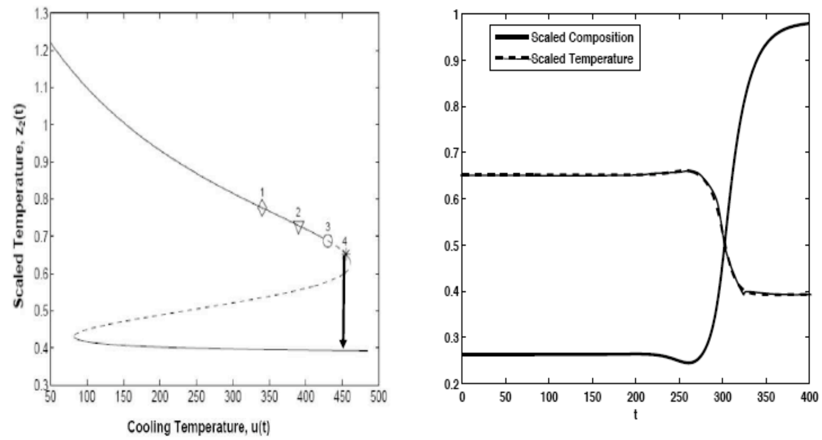
Monomer

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

Grafting sites

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

Polymer Reactor - Unstable Steady State



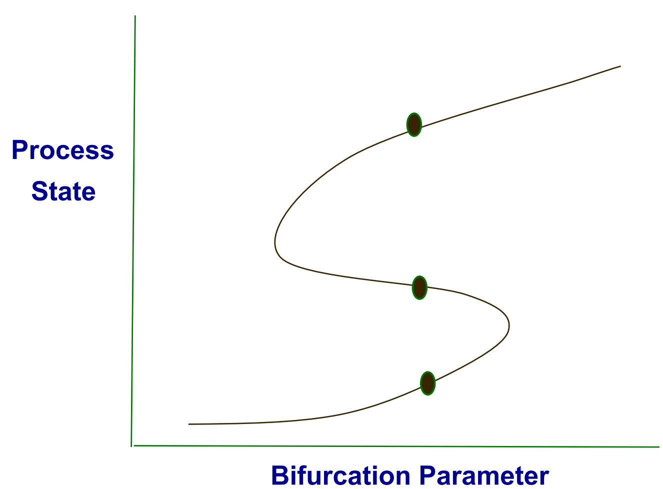
CSTR steady state cannot be maintained without stabilization

Drift to another steady state with sequential approach

65

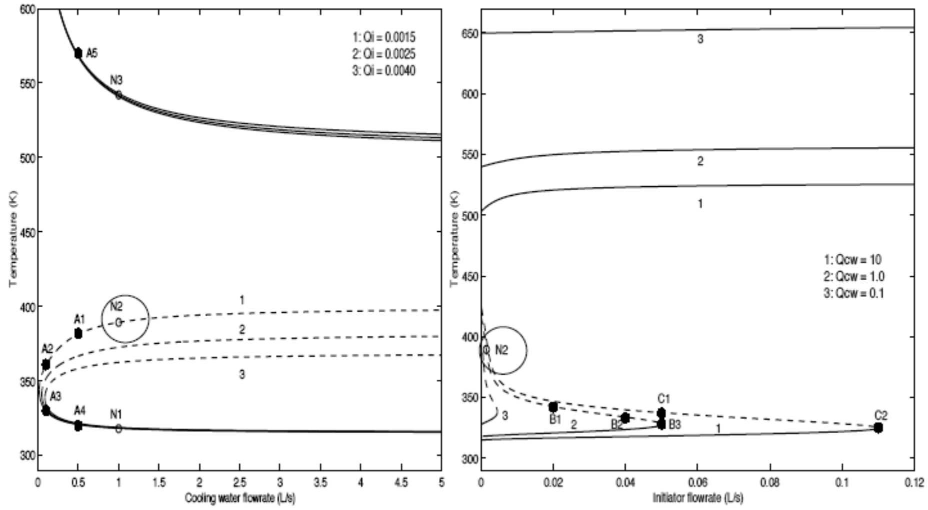
Phase Diagram of Steady States

Transitions considered among all steady states



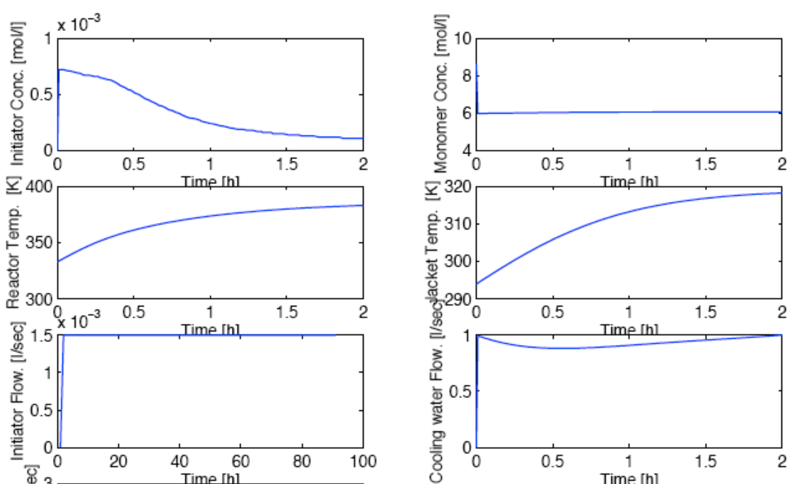
66

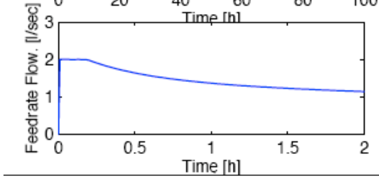
Phase Diagram of Steady States



67

Startup to Unstable Steady State

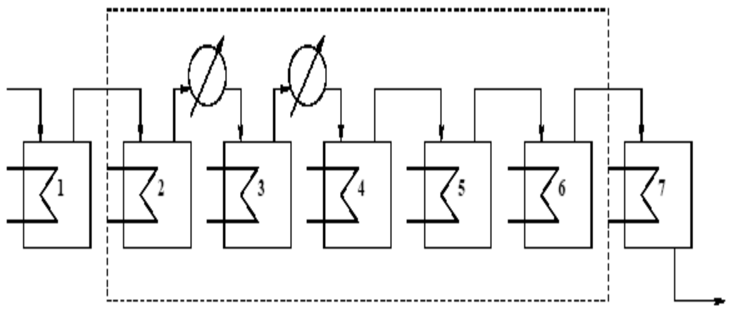




- 926 variables
- 476 constraints
- 36 iters. / 0.95 CPU s (P4)

68

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.

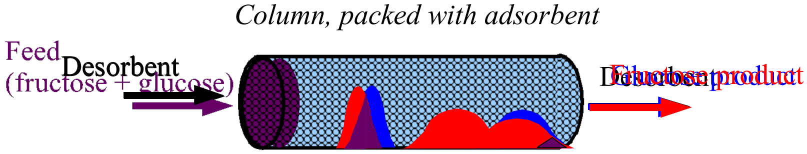
69

Simulated Moving Beds
(Kawajiri, B., 2005, 2006)

Sequential batch process,

making use of difference in affinity to the adsorbent

Column, packed with adsorbent



Two components separates as moving toward the end
(Difference in affinity)

70

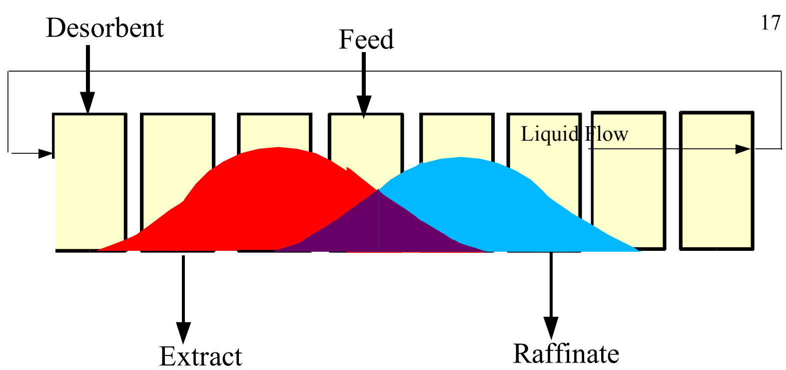
SMB Applications

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

Separate 'good' from 'bad' compounds based on chirality

Step

Cyclic Steady State



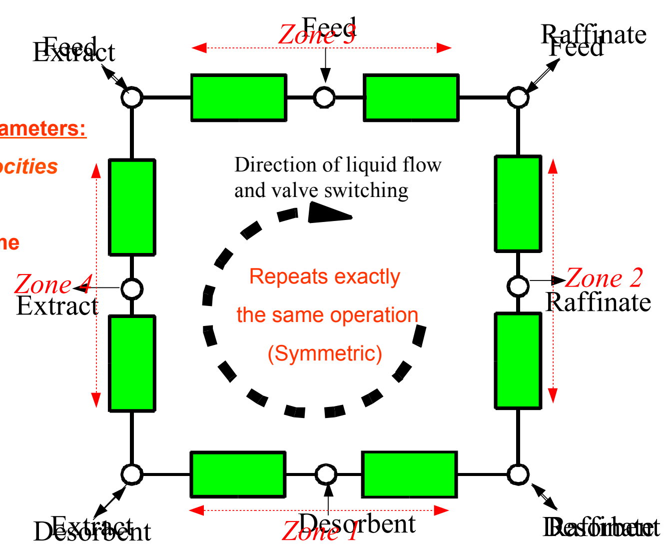
Simulated Moving Bed

Operating parameters:

4 Zone velocities

+

Step time



Formulation of Optimization Problem

Zone velocities — Step time

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

(Maximize average feed velocity)

Product requirements

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

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

Bounds on liquid velocities

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

SMB model

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

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

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

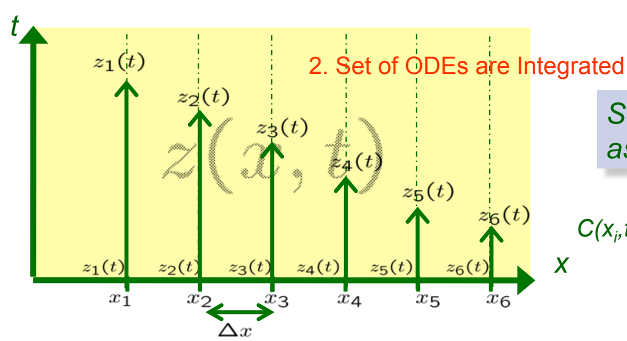
CSS constraint

$$C_i(x,t_0) = C_{i+1}(x,t_0+t_{step})$$

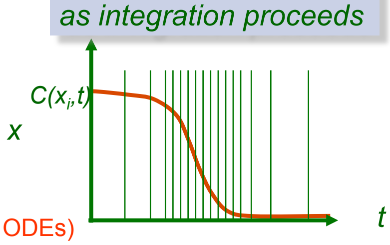
$$q_i(x,t_0) = q_{i+1}(x,t_0+t_{step})$$

73

Treatment of PDEs: Single Discretization



Step size determined as integration proceeds



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

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

$subject\ \ to:$

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

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

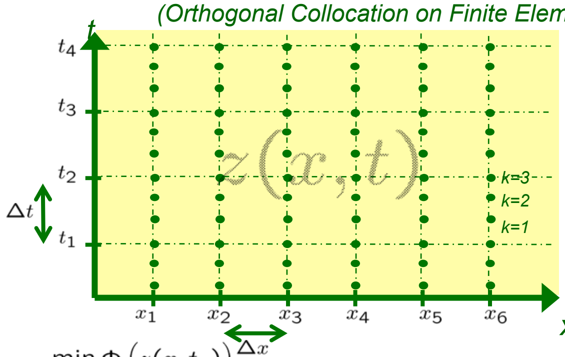
$$h\left(p, z(x,t)\right) = 0$$

PDE → ODE (Handled by integrator)

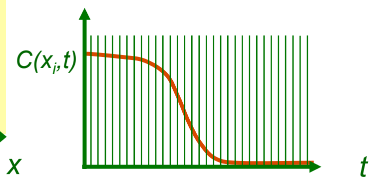
74

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:

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

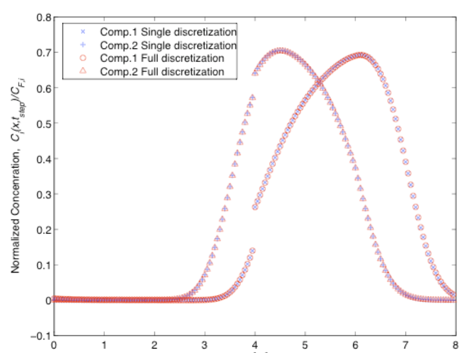
$$g\left(p, z(x,t)\right) \le 0 \qquad z_{i,j,k} = z^0_{i,j} + h_j \sum_{k=1}^{3} \Omega_{j,k} \left(\frac{dz}{dt}\right)_{i,j,k}$$

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

PDE → Algebraic equations

75

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

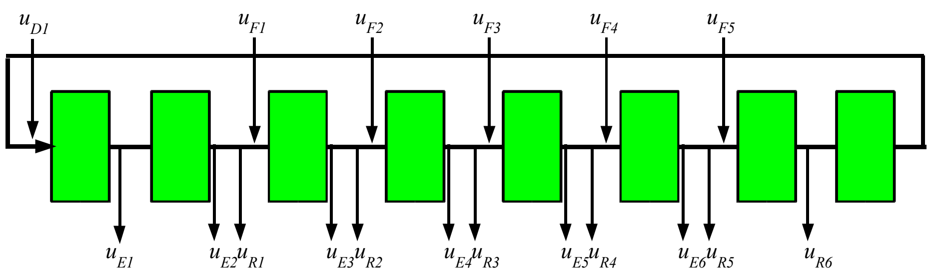
Optimal feed velocity: 0.52 m/h

*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

76

Superstructure for Zone Configuration



<u>Constraints:</u>

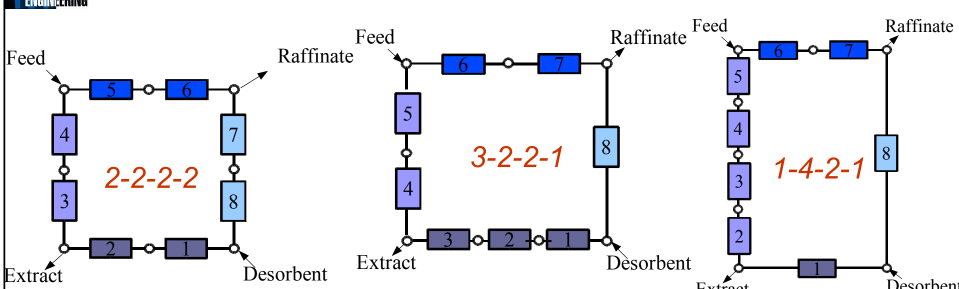
All velocities are constant

8 columns, multiple streams

Repeat exactly same stream policies for each step

77

Allows straightforward zone configuration optimization



Optimal feed velocity: 0.906 m/h　　0.957 m/h　　1.052 m/h

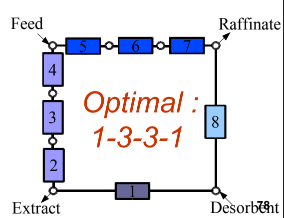
# of columns	8	10	12	14
# of zone configurations	35	84	165	286

Too many configurations to enumerate!

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

CPU Time: 1.73 min*

Optimal feed velocity: 1.158 m/h

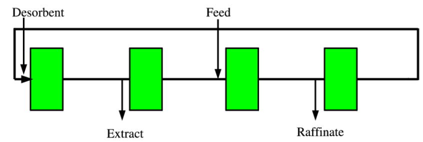


**On Xeon 3.2GHz*

78

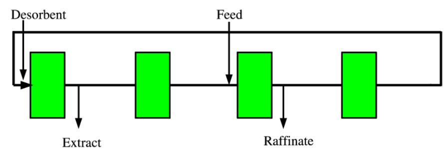
Nonstandard SMB: Addressed by Extended Superstructure NLP

- **Standard SMB**



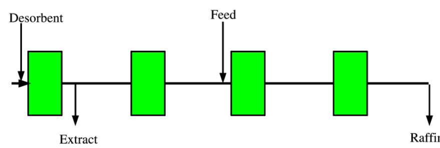
- **VARICOL**

(Asynchronous switching)

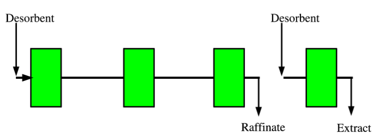


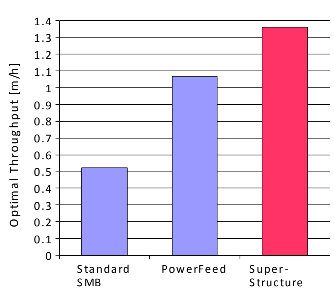
- **Three Zone**

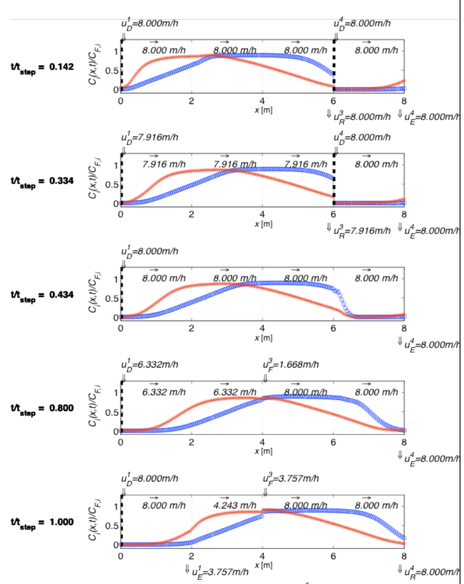
(Circulation loop is cut open)



Optimal Operating Scheme:
Result of Superstructure Optimization





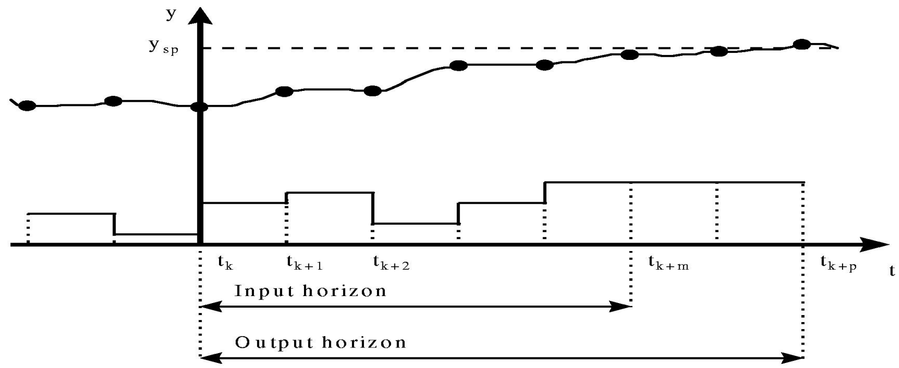


CPU Time for optimization: 9.03 min*

34098 variables, 34013 equations

*on Xeon 3.2 GHz

Nonlinear Model Predictive Control (NMPC)



$$\min_{u} \sum_{j} \| y(t_{k+j}) - y^{sp} \|^2_{Q_y} + \sum_{j} \| \boldsymbol{u}(t_{k+j}) - \boldsymbol{u}(t_{k+j-1}) \|^2_{Q_u}$$

$$s.t. \quad z'(t) = F(z(t), y(t), \boldsymbol{u}(t), t)$$

$$0 = G(z(t), y(t), \boldsymbol{u}(t), t)$$

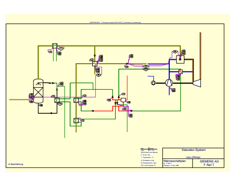
$$z(t) = z(t_k)$$

Bound Constraints

Other Constraints

Dynamic optimization in a MATLAB Framework

Dynamic Process



Dynamic Optimization Problem

Process Model

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

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

Inequality Constraints

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

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

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

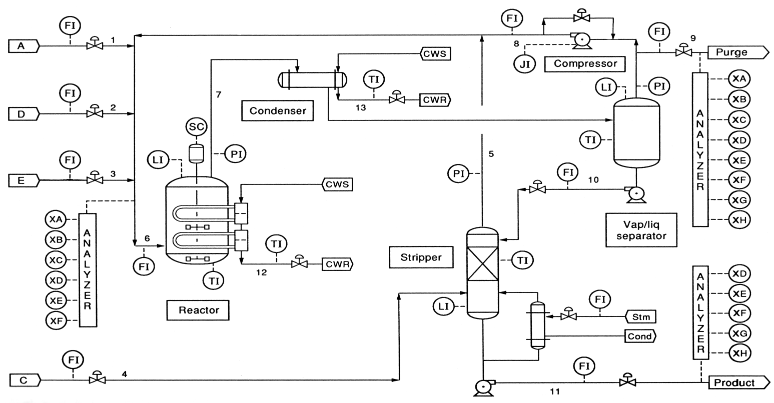
Constraints at Final Time

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

Objective Function

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

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

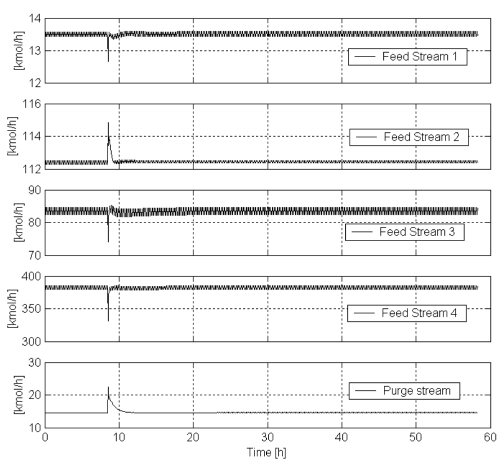
Large-scale Sparse block-diagonal NLP

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]

Case Study:
Change Reactor pressure by 60 kPa



Control profiles

All profiles return to their base case values

Same production rate

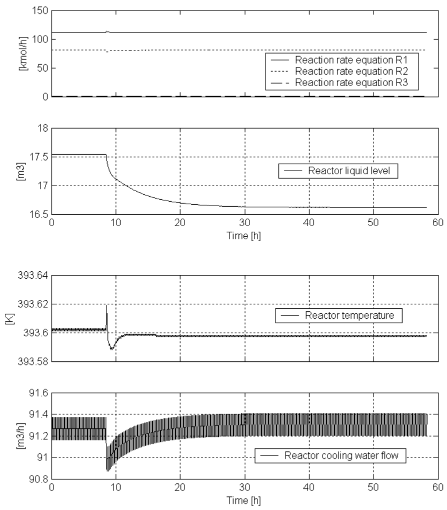
Same product quality

Same control profile

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

Less compressor load

TE Case Study – Results I



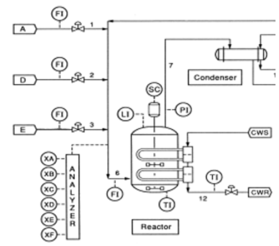
Shift in TE process

Same production rate

More volume for reaction

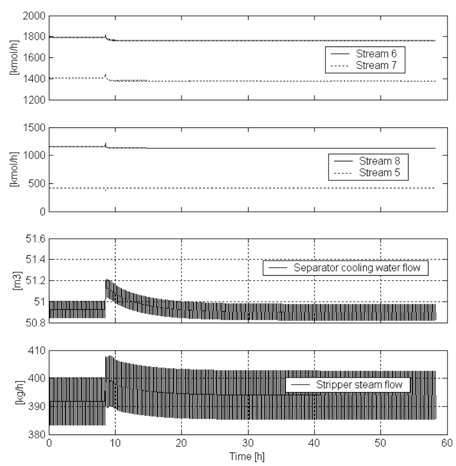
Same reactor temperature

Initially less cooling water flow (more evaporation)



87

Case Study- Results II



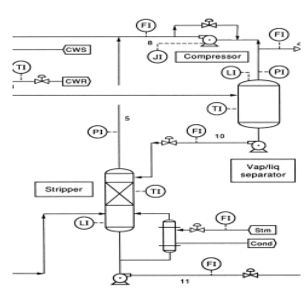
Shift in TE process

Shift in reactor effluent to more condensables

Increase cooling water flow

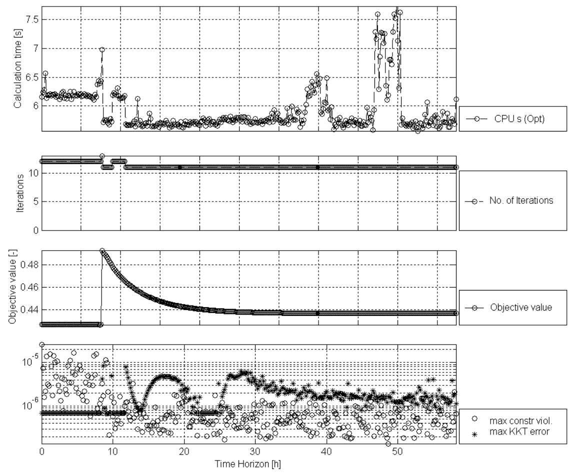
Increase stripper steam to ensure same purity

Less compressor work



88

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

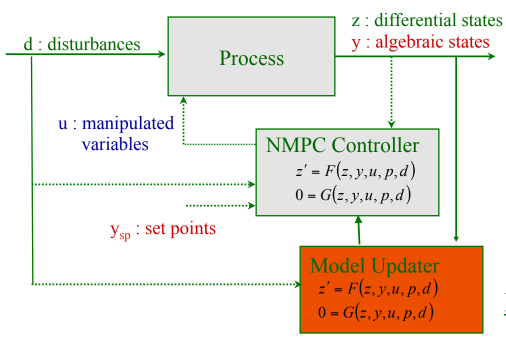
Could not be solved within sampling times

> 100 Iterations

89

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

$$\min_p \sum_j \| y(t_{k-j}) - \hat{y}(t_{k-j}) \|^2_{Q_y} + \sum \| p^k \|^2_{Q_u}$$

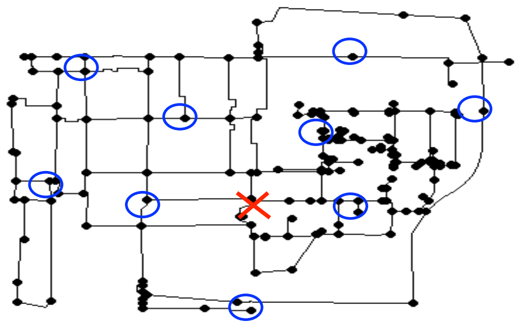
$$\text{s.t.} \quad \begin{aligned} & z'(t) = F(z(t), y(t), \mathbf{p}, u(t), t) \\ & 0 = G(z(t), y(t), \mathbf{p}, u(t), t) \\ & z(t) = z_{k-N} \end{aligned}$$

Bound Constraints

Other Constraints

90

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

91

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

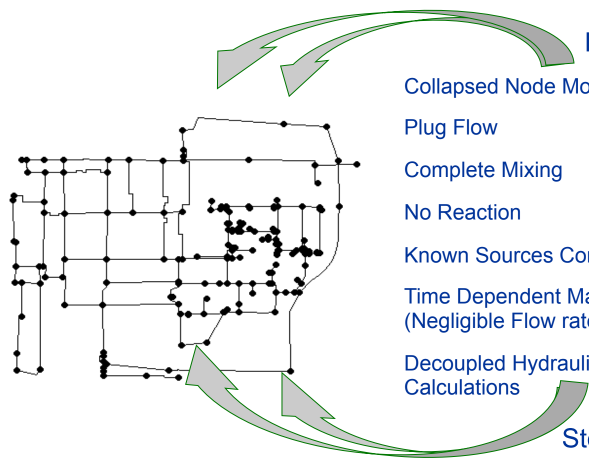
$$\left.\begin{aligned} V_k(t)\frac{d\hat{c}_k(t)}{dt} &= \left(\sum_{i\in\Gamma_k(t)} Q_i(t)\,\bar{c}_i(x{=}\mathcal{O}_i(t),t)\right) + m_k(t) - \left[\left(\sum_{i\in\Gamma_k(t)} Q_i(t)\right) + Q_k^{ext}(t) + Q_k^{inj}(t)\right]\hat{c}_k(t), \\ \hat{c}_k(t{=}0) &= 0, \end{aligned}\right\} \forall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

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

Injection Terms Only

92

Water Quality Model



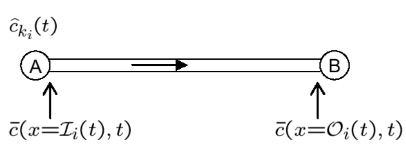
Pipes, Valves, Pumps

- Collapsed Node Models
- Plug Flow
- Complete Mixing
- No Reaction
- Known Sources Contaminant Free
- Time Dependent Mass Injections at All Nodes (Negligible Flow rates)
- Decoupled Hydraulics and Water Quality Calculations

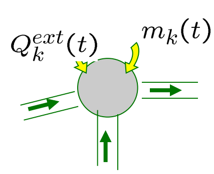
Storage Tanks, Junctions

93

Modeling Water Distribution Systems



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



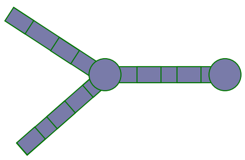
$$\hat{c}_k(t) = \frac{\left(\sum_{i\in\Gamma_k(t)} Q_i(t)\,\bar{c}_i(x{=}\mathcal{O}_i(t),t)\right) + m_k(t)}{\left(\sum_{i\in\Gamma_k(t)} Q_i(t)\right) + Q_k^{ext}(t) + Q_k^{inj}(t)},$$

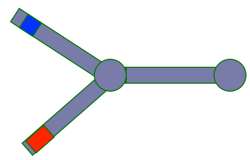
$$V_k(t)\frac{d\hat{c}_k(t)}{dt} = \left(\sum_{i\in\Gamma_k(t)} Q_i(t)\,\bar{c}_i(x{=}\mathcal{O}_i(t),t)\right) + m_k(t) - \left[\left(\sum_{i\in\Gamma_k(t)} Q_i(t)\right) + Q_k^{ext}(t) + Q_k^{inj}(t)\right]\hat{c}_k(t),$$
$$\hat{c}_k(t{=}0) = 0,$$

94

Existing Simulation Techniques

Eulerian	Lagrangian
Discretize in time and space	Discretize in time alone
Track concentration at fixed points or volumes	Track concentration of elements as they move
Local process for simulation, but global treatment needed for simultaneous optimization	Algorithmic in nature
Same as our Discretization	No Straightforward Representation
Too Many Constraints	Derivative Calculations?



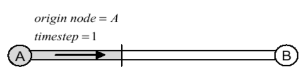


Review of these methods by Rossman and Boulos, 1996.

95

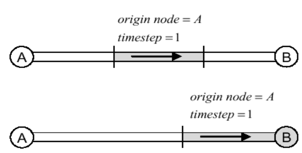
Origin Tracking Algorithm

$l=1$



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

$l=5$



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

Known Hydraulics – Function of Time

Pipe Network PDEs Linear in Concentration

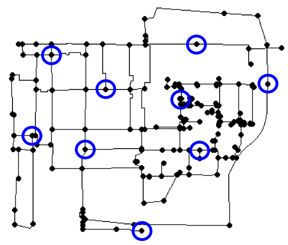
Pipe by Pipe PDEs

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

Removes Need to Discretize in Space

96

Source Inversion Formulation

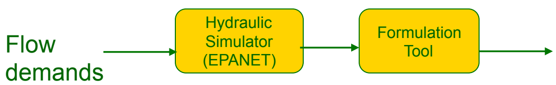


Water network demands known.

Linear PDEs for concentration

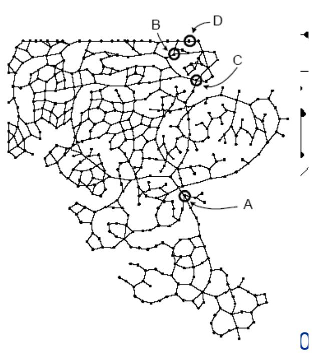
Convert to delay-differential equations

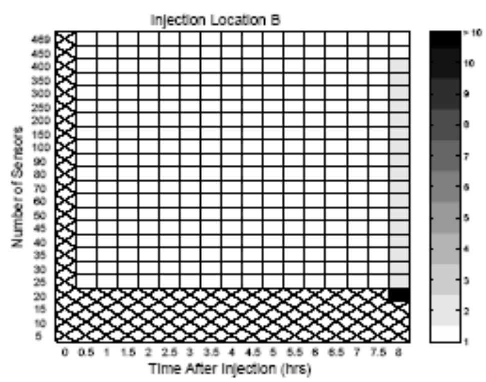
Apply implicit R-K formulation



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

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)

Dynamic Optimization Problems:
A Closer Look at Simultaneous Methods
(S. Kameswaran)

$$\begin{aligned}
\min_{u(t),p,t_f} \quad & \varphi\left(z(t),y(t),u(t),p,t_f\right) \\
\text{s.t.} \quad & \frac{d}{dt}z(t) = f\left(z(t),y(t),u(t),p\right); \quad z(0)=z_0 \\
& c\left(z(t),y(t),u(t),p\right)=0 \\
& g\left(z(t),y(t),u(t),p\right)\le 0 \\
& g_f\left(z(t_f)\right)\le 0
\end{aligned}$$

Optimize then discretize

Variational / Indirect Approach

Discretize MPBVP

Discretize then Optimize

Use NLP Solver / Direct Approach

Direct Simultaneous Approach

Multiple Shooting

Sequential

L.T. Biegler
J.T. Betts
J. F. Bonnans
W. Hager

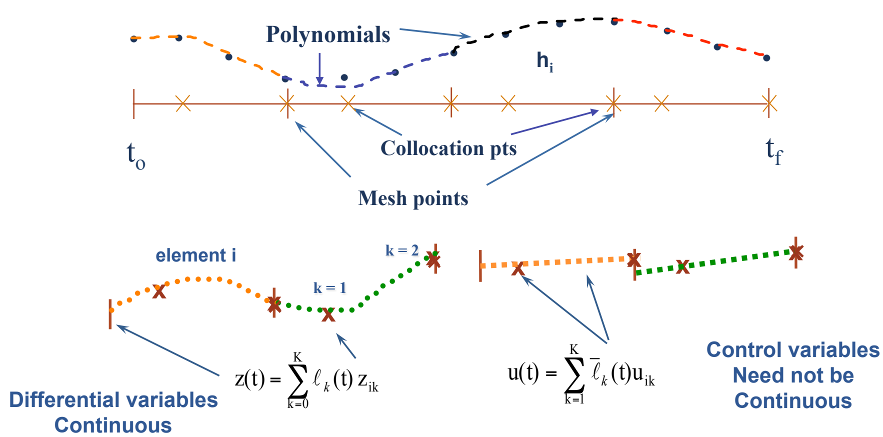
H.G. Bock

R.W.H. Sargent
C.C. Pantelides
P.I. Barton
W. Marquardt

99

Orthogonal Collocation

- Orthogonal collocation is one particular discretization method



- Radau and Gauss Collocations – Choice of interpolation points

100

Are we doing the right thing ?
Second Order Explicit Runge-Kutta Discretization

- **Example – Hager (1976, 2001)**

$$\min_{u(t)} \quad \frac{1}{2}\int_{t=0}^{1}\left(2z^2(t)+u^2(t)\right)dt$$
$$\text{s.t.} \quad \frac{dz(t)}{dt}=\frac{1}{2}z(t)+u(t), \qquad z(0)=1$$

$$z^*(t)=\frac{2e^{3t}+e^3}{e^{\frac{3}{2}t}\left(2+e^3\right)}$$
$$u^*(t)=\frac{2\left(e^{3t}-e^3\right)}{e^{\frac{3}{2}t}\left(2+e^3\right)}$$

- **Second Order Explicit Runge-Kutta method for discretization**

$$\min \quad \frac{h}{2}\sum_{k=0}^{N-1}\left(2z^2_{k+\frac{1}{2}}+u^2_{k+\frac{1}{2}}\right)dt$$
$$\text{s.t.} \quad z_{k+\frac{1}{2}}=z_k+\frac{h}{2}\left(\frac{1}{2}z_k+u_k\right); \; k=0,\ldots,N-1$$
$$z_{k+1}=z_k+h\left(\frac{1}{2}z_{k+\frac{1}{2}}+u_{k+\frac{1}{2}}\right); \; k=0,\ldots,N-1$$
$$z_0=1$$

101

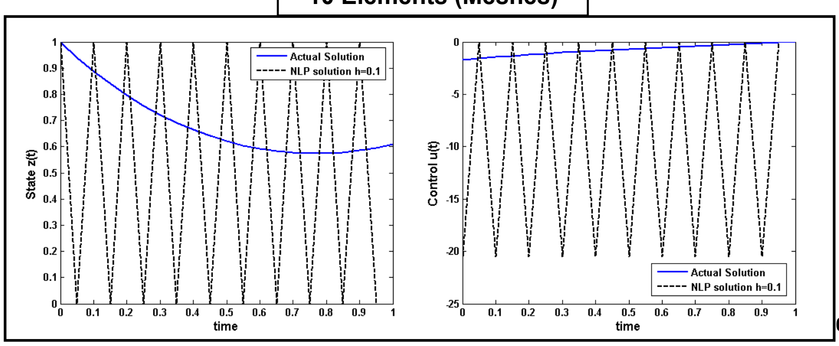
Are we doing the right thing ?
Second Order Explicit Runge-Kutta Discretization

- **Example – Hager (1976, 2001)**

$$\min_{u(t)} \quad \frac{1}{2}\int_{t=0}^{1}\left(2z^2(t)+u^2(t)\right)dt$$
$$\text{s.t.} \quad \frac{dz(t)}{dt}=\frac{1}{2}z(t)+u(t), \qquad z(0)=1$$

$$z^*(t)=\frac{2e^{3t}+e^3}{e^{\frac{3}{2}t}\left(2+e^3\right)}$$
$$u^*(t)=\frac{2\left(e^{3t}-e^3\right)}{e^{\frac{3}{2}t}\left(2+e^3\right)}$$

10 Elements (Meshes)



02

Are we doing the right thing ?
Second Order Explicit Runge-Kutta Discretization

- **Example – Hager (1976, 2001)**

$$\min_{u(t)} \quad \frac{1}{2}\int_{t=0}^{1}\left(2z^2(t)+u^2(t)\right)dt$$
$$\text{s.t.} \quad \frac{dz(t)}{dt} = \frac{1}{2}z(t)+u(t), \qquad z(0)=1$$

$$z^*(t) = \frac{2e^{3t}+e^3}{e^{\frac{3}{2}t}\left(2+e^3\right)}$$
$$u^*(t) = \frac{2\left(e^{3t}-e^3\right)}{e^{\frac{3}{2}t}\left(2+e^3\right)}$$

NLP solution $\longrightarrow$ $h \rightarrow 0$, $N = \dfrac{t_f - t_0}{h} \rightarrow \infty$ $\longrightarrow$ **Solution of the Original Problem**

DOES NOT CONVERGE !

- **Discretization scheme is a crucial component for success**
- **Extremely important to know if the solution obtained is correct**
- **Literature Review:** Hager (1976,2001), Polak (1996), Malanowski and Maurer (1997) – restricted to convex problems (or) just consistency (or) results only at selected points

Convergence Properties for Gauss Collocation

(P)
$$\min_{z(t),u(t)} \quad \varphi(z(t_f))$$
$$\text{s.t.} \quad \frac{dz(t)}{dt} = f(z(t),u(t)), \qquad z(t_0)=z_0$$
$$g_f(z(t_f)) = 0$$

Direct Simultaneous Approach

$\min \varphi(z_f)$

s.t. Discretized ODEs

$+$

Final-time Constraints

Step Size h

Indirect Approach

Optimality Conditions for (P)

- DAE system

Discretization

Stepsize h

NLP solution ? $\longrightarrow$ $h \rightarrow 0$, $N = \dfrac{t_f - t_0}{h} \rightarrow \infty$ $\longrightarrow$ **Convergence to OCP**

$$\min \ \Phi(z(t_f))$$
$$s.t. \ \frac{dz}{dt} = f(z(t), u(t)), \quad z(t_0) = z_0$$
$$h_E(z(t_f)) = 0$$

$$\frac{dz}{dt} = f(z, u), \quad z(t_0) = z_0$$
$$h_E(z(t_f)) = 0$$
$$\frac{d\lambda}{dt} = -\frac{\partial f(z, u)}{\partial z}\lambda(t)$$
$$\lambda(t_f) = \frac{\partial \Phi(z(t_f))}{\partial z} + \frac{\partial h_E(z(t_f))}{\partial z}\eta_E$$
$$\frac{\partial f(z, u)}{\partial u}\lambda = 0$$

105

$$\min \ \Phi(z(t_f))$$
$$s.t. \ \frac{dz}{dt} = f(z(t), u(t)), \quad z(t_0) = z_0$$
$$h_E(z(t_f)) = 0$$

$$Min \ \Phi(z_f)$$
$$s.t. \ \sum_{j=0}^{K} \dot{\ell}_j(\tau_k) z_{ij} - h_i f(z_{ik}, u_{ik}) = 0$$
$$k \in \{1, \ldots, K\}, \ i \in \{1, \ldots, N\}$$
$$z_{i+1,0} = \sum_{j=0}^{K} \ell_j(1) z_{ij}, \quad i = 1, \ldots, N-1$$
$$z_f = \sum_{j=0}^{K} \ell_j(1) z_{Nj}, \quad z_{1,0} = z(t_0)$$
$$h_E(z_f) = 0,$$

106

$$\mathcal{L} = \Phi(z_f) + \eta_E^T h_E(z_f) + \sum_{i=1}^{N}\sum_{j=1}^{K}\left\{\bar{\lambda}_{ij}^T\left[h_i f(z_{ij}, u_{ij}) - \sum_{k=0}^{K}\dot{\ell}_k(\tau_j) z_{ik}\right]\right\}$$
$$+ \sum_{i=1}^{N-1}\bar{\nu}_i^T(z_{i+1,0} - \sum_{j=0}^{K}\ell_j(1) z_{ij}) + \bar{\nu}_0^T(z_{1,0} - z(t_0)) + \bar{\nu}_N^T(z_f - \sum_{j=0}^{K}\ell_j(1) z_{N,j}).$$

$$\omega_j \lambda_{ij} = \bar{\lambda}_{ij},\ \omega_j > 0$$

$$\nabla_{z_f}\mathcal{L} = \nabla_z \Phi(z_f) + \nabla_z h_E(z_f)\eta_E + \bar{\nu}_N = 0$$
$$\nabla_{z_{ij}}\mathcal{L} = \omega_j h_i \nabla_z f(z_{ij}, u_{ij})\lambda_{ij} - \sum_{k=1}^{K}\omega_k \lambda_{ik}\dot{\ell}_j(\tau_k) - \bar{\nu}_i \ell_j(1) = 0$$
$$\nabla_{u_{ij}}\mathcal{L} = \omega_j h_i \nabla_u f(z_{ij} u_{ij})\lambda_{ij} = 0$$
$$\nabla_{z_{i,0}}\mathcal{L} = \bar{\nu}_{i-1} - \bar{\nu}_i \ell_0(1) - \sum_{k=1}^{K}\omega_k \lambda_{ik}\dot{\ell}_0(\tau_k) = 0$$

107

$$\sum_{k=1}^{K}\omega_k \lambda_{ik}\dot{\ell}_j(\tau_k) = \int_0^1 \lambda_i(\tau)\dot{\ell}_j(\tau)d\tau$$
$$= \lambda_i(1)\ell_j(1) - \lambda_i(0)\ell_j(0) - \int_0^1 \dot{\lambda}_i(\tau)\ell_j(\tau)d\tau$$
$$= \lambda_i(1)\ell_j(1) - \lambda_i(0)\ell_j(0) - \sum_{k=1}^{K}\omega_k \dot{\lambda}_i(\tau_k)\ell_j(\tau_k)$$

$$\sum_{k=1}^{K}\omega_k \lambda_{ik}\dot{\ell}_j(\tau_k) = \lambda_i(1)\ell_j(1) - \omega_j \dot{\lambda}_i(\tau_j),\ j = 1, \ldots, K$$
$$\sum_{k=1}^{K}\omega_k \lambda_{ik}\dot{\ell}_0(\tau_k) = \lambda_i(1)\ell_0(1) - \lambda_i(0)$$

Substituting these relations into (10.32b) and (10.32d) leads to:

$$\nabla_{z_{ij}}\mathcal{L} = \omega_j[h_i(\nabla_z f(z_{ij}, u_{ij})\lambda_{ij} + \dot{\lambda}_i(\tau_j)] - (\bar{\nu}_i + \lambda_i(1))\ell_j(1) = 0$$
$$\nabla_{z_{i,0}}\mathcal{L} = \lambda_i(0) + \bar{\nu}_{i-1} - (\bar{\nu}_i + \lambda_i(1))\ell_0(1) = 0$$

108

Consistent Optimality Conditions

$$\frac{dz}{dt} = f(z,u), \quad z(t_0) = z_0$$

$$h_E(z(t_f)) = 0$$

$$\frac{d\lambda}{dt} = -\frac{\partial f(z,u)}{\partial z}\lambda(t)$$

$$\lambda(t_f) = \frac{\partial \Phi(z(t_f))}{\partial z} + \frac{\partial h_E(z(t_f))}{\partial z}\eta_E$$

$$\frac{\partial f(z,u)}{\partial u}\lambda = 0$$

$$\nabla_{z_f}\mathcal{L} = \nabla_{z_f}\Phi + \nabla_{z_f} h_E \eta_E - \lambda_N(1) = 0$$

$$\nabla_{z_{ij}}\mathcal{L} = \omega_j[\dot{\lambda}_i(\tau_j) + h_i \nabla_z f(z_{ij}, u_{ij})\lambda_{ij}] = 0$$

$$\nabla_{u_{ij}}\mathcal{L} = \omega_j h_i \nabla_u f(z_{ij}, u_{ij})\lambda_{ij} = 0$$

$$\nabla_{z_{i,0}}\mathcal{L} = \lambda_{i-1}(1) - \lambda_i(0) = 0$$

109

Convergence Rates – Runge-Kutta Methods

Indirect Approach $\longleftrightarrow$ True Solution of Optimal Control Problem (P)

Numerical Methods for BVP/DAEs

- Symmetric Implicit R-K methods and Gauss collocation –

True Solution of Optimal Control Problem (P)

$\updownarrow$

Necessary Conditions NLP $\longleftrightarrow$ Indirect Approach

- Radau collocation
  - Highest Precision after Gauss Collocation
  - Better Stability Properties than Gauss

Necessary Conditions NLP $\not\longleftrightarrow$ Indirect Approach

110

Convergence Rates Unconstrained Optimal Control Problems

Tool : Mean Value theorem for vector functions of vector variables.

$$r = \phi(w^*) - \underbrace{\phi(w)}_{\mathbf{0}} = J(w^* - w) = J\Delta w \;\; = O(h^K)$$

$$\text{where } J = \int_0^1 \phi'(\zeta w^* + (1-\zeta)w)d\zeta$$

$\phi(.)$ - **Optimality Conditions for the NLP (KKT conditions)**

w - **NLP solution (primal variables and Lagrange multipliers)**

w^* - **Sampled true solution (states, controls and multipliers)**

J - **KKT matrix**

$\hat{A}$ - **Jacobian matrix of Collocation Equations – need nonsingular KKT matrix**

$$J = \begin{pmatrix} H & \hat{A}^T \\ \hat{A} & 0 \end{pmatrix}; \quad \hat{A} = \begin{bmatrix} \hat{C} & N \end{bmatrix}; \quad H \approx \text{Hessian of the Lagrangian}$$

CAUTION – finer meshes imply increase in linear system

Convergence Rates for Unconstrained OCPs - 2

$$J(w^* - w) = J\Delta w = \begin{pmatrix} H & \hat{A}^T \\ \hat{A} & 0 \end{pmatrix} \begin{pmatrix} \Delta \mathrm{w_x} \\ \Delta \mathrm{w_d} \\ \Delta \mathrm{w}_\eta \end{pmatrix} = r$$

Linear Transformation
"Range and Null Space Decomposition"

$\hat{A}Z = 0; \quad [Y \mid Z]$ is nonsingular

$$\begin{pmatrix} \Delta \mathrm{w_x} \\ \Delta \mathrm{w_d} \end{pmatrix} = [\mathrm{Y} \mid \mathrm{Z}] \begin{pmatrix} \mathrm{p}_Y \\ \mathrm{p}_Z \end{pmatrix}$$

Convergence Rates for Unconstrained OCPs - 3

- Solve Linear System to determine convergence properties

$$\begin{pmatrix} Y^T H Y & Y^T H Z & \hat{C}^T \\ Z^T H Y & Z^T H Z & 0 \\ \hat{C} & 0 & 0 \end{pmatrix} \begin{pmatrix} p_Y \\ p_Z \\ \Delta w_\eta \end{pmatrix} = \begin{pmatrix} r_x \\ r_d - N^T \hat{C}^{-T} r_x \\ r_\eta \end{pmatrix}$$

- Assumptions (can be relaxed easily)
  - Equally spaced meshes
  - NLP solution (primal and dual) does not diverge as *h→0*
- **Theorem:** $\left\|\hat{C}^{-1}\right\|_\infty = O\left(\frac{1}{h}\right)$ and $\left\|\hat{C}^{-1}\right\|_1 = O\left(\frac{1}{h}\right)$
- **Assumption:** $\left\|\left(Z^T H Z\right)^{-1}\right\|_\infty = O\left(\frac{1}{h}\right)$
  - Diagonally dominant red. Hess.
  - Systems satisfying Coercivity (Dontchev and Hager, 1998)
  - Numerical experiments

REGULAR (or Nonsingular) OCPs

113

Radau Convergence Results

- Convergence for states, control, adjoints/multipliers (scaled by ω_j) at the rate of $O(h^K)$
- Requires regularity assumptions on KKT matrix

$$\begin{aligned} \text{Error in the states} &\le C_1 h^K \\ \text{Error in the controls} &\le C_2 h^K \\ \text{Error in the adjoints} &\le C_3 h^K \end{aligned}$$

- Analysis extended to include final-time equality constraints

$$\begin{bmatrix} \mathcal{H} & \hat{\mathcal{A}} & C_s \\ \hat{\mathcal{A}}^T & 0 & 0 \\ C_s^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta w_z \\ \Delta w_\lambda \\ \Delta w_{\mu_f} \end{bmatrix} = \begin{bmatrix} r_\lambda \\ r_z \\ 0 \end{bmatrix}$$

Controllability of the associated LTV system → regularity of expanded KKT matrix and same order properties

114

Regularity Assumptions

- Equally spaced meshes – $h = \frac{t_f - t_0}{NE}$
- The NLP solution (primal and dual) does not diverge from the "true" continuous-time solution as h→ *0*.

$$J = \begin{pmatrix} H & \hat{A}^T \\ \hat{A} & 0 \end{pmatrix} \qquad \left.\begin{array}{l} \hat{A} \text{ - Full row rank} \\ Z^T HZ \text{ - Positive Definite} \end{array}\right\} \; J \text{ - invertible}$$

- Matrix sizes increase with *h*, and this needs to be handled appropriately.
- Requires linearly independent constraint gradients
  - Guaranteed with ODEs and index 1 DAEs
  - High index models require further analysis, open questions, see Betts, Campbell example…
- Require $\left\| (Z^T HZ)^{-1} \right\| = O(h^{-1})$

  Applies to Regular (Nonsingular Control Problems)

115

What about choosing element length, h_i?
Mesh placement heuristics

⇒For a sufficient number of elements
An algorithm that guarantees no changes in the active set within each element is consistent with discrete variational conditions

⇒Adjoint variables approximated as piecewise polynomials in each element from multipliers

⇒Still need to check for sufficient number of elements
⇒Error control
⇒Constant Hamiltonian

Dynamic Optimization Problem

$$\min \; \Phi\left(z(t_f)\right)$$

s.t.

$$\boxed{\begin{aligned} &\frac{dz(t)}{dt} = f\left(z(t), y(t), u(t), t\right) \\ &g\left(z(t), y(t), u(t), t, p\right) = 0 \end{aligned}}$$

Index 1 DAE

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

$$\cancel{z^l \le z(t) \le z^u}$$

$$\cancel{y^l \le y(t) \le y^u}$$

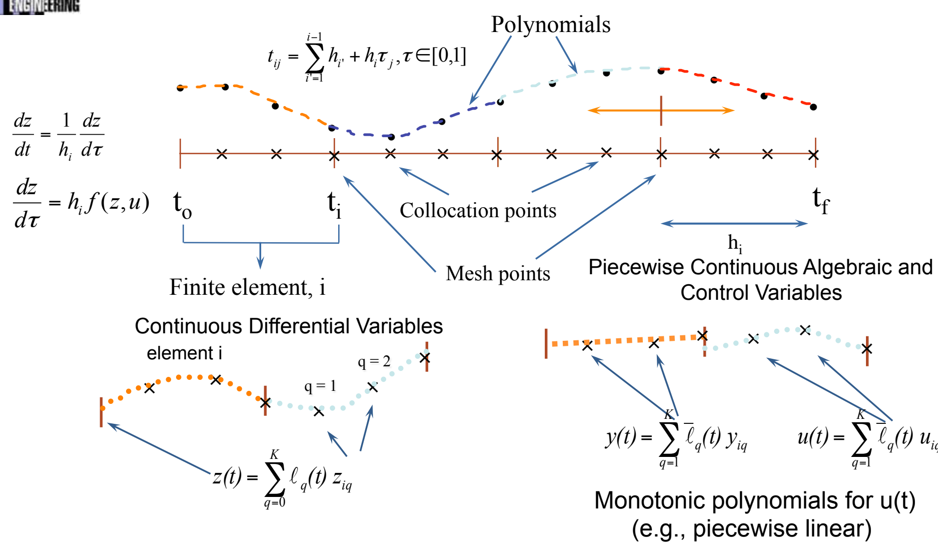
$$u^l \le u(t) \le u^u$$

t, time
z, differential variables
y, algebraic variables

t_f, final time
u, control variables

117

Radau Collocation on Finite Elements



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

Nonlinear Programming Problem

$$\min \; \Phi(z_f)$$

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

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

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

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

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

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

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

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

Use finite elements, h_i, as variables to enforce accurate solution and determine break points for $u(t)$.

- Add $0 \le h_i \le h_u$, $\Sigma\, h_i = t_f$
- Add $-\varepsilon \le T(h, z, u) \le \varepsilon$ to enforce state error

119

Properties of Gauss/Radau Direct Transcription

Stability and Accuracy of Orthogonal Collocation

- Equivalent to performing a *fully implicit* Runge-Kutta integration of the DAE models at Gaussian (Radau) points
- Superconvergence ($O(h^{2K})$ or ($O(h^{2K-1})$) at τ = 1 for K collocation pts.
- State Variable Error Criteria (Russell and Christensen, 1978)

$$max_{t\in[0,t_f]} \|e(t)\| \le C_1 max_{i\in\{1,\dots,N\}}(max_{t\in[t_{i-1},t_i]} \|T_i(t)\|)$$

$$T_i(t_{i,nc}) = \begin{bmatrix} \frac{dz^K(t)}{d\tau} - h_i f(z^K(t_{i,nc}), y^K(z(t_{i,nc})), u^K(t_{i,nc})) \\ g(z^K(t_{i,nc}), y^K(z(t_{i,nc})), u^K(t_{i,nc})) \end{bmatrix}$$

Analysis of the Optimality Conditions

- Equivalence between NLP KKT and discretized variational necessary conditions
- Convergence Rate for Radau discretization (Kameswaran, B. 2008)
    - Error in States = $O(h^K)$
    - Error in Adjoints = $O(h^K)$
    - Error in Controls = $O(h^K)$

Elements of MFE Formulation

$$
\begin{aligned}
\min \quad & \Phi(z_f) \\
s.t. \quad & \sum_{j=0}^{K} \dot{\ell}_j(\tau_k) z_{ij} - h_i f(z_{ik}, y_{ik}, u_{ik}) = 0 \\
& g(z_{ik}, y_{ik}, u_{ik}) = 0 \\
& u_L \le u_{ik} \le u_U, \quad u_{ik} = \sigma(\tau_k, v_i) \\
& k \in \{1, \ldots, K\},\ i \in \{1, \ldots, N\} \\
& z_{i+1,0} = \sum_{j=0}^{K} \ell_j(1) z_{ij}, \quad i = 1, \ldots, N-1 \\
& z_f = \sum_{j=0}^{K} \ell_j(1) z_{Nj}, \quad z_{1,0} = z(t_0) \\
& \psi(z_f) \le 0
\end{aligned}
$$

Elements of MFE Formulation

$$
\begin{aligned}
\min \quad & \Phi(z_f) \\
s.t. \quad & \sum_{j=0}^{K} \dot{\ell}_j(\tau_k) z_{ij} - \boxed{h_i} f(z_{ik}, y_{ik}, u_{ik}) = 0 \\
& g(z_{ik}, y_{ik}, u_{ik}) = 0 \\
& u_L \le u_{ik} \le u_U, \quad u_{ik} = \sigma(\tau_k, v_i) \\
& k \in \{1, \ldots, K\},\ i \in \{1, \ldots, N\} \\
& z_{i+1,0} = \sum_{j=0}^{K} \ell_j(1) z_{ij}, \quad i = 1, \ldots, N-1 \\
& z_f = \sum_{j=0}^{K} \ell_j(1) z_{Nj}, \quad z_{1,0} = z(t_0) \\
& \psi(z_f) \le 0 \\
& -\varepsilon \le \bar{C}\, T_i(t_{i,nc}) \le \varepsilon \\
& 0 \le h_i \le t_f, \ \sum_{i=1}^{N} h_i = t_f.
\end{aligned}
$$

<u>Solution Strategy Concerns</u>

- What is N? Too few elements leads to suboptimal solution
- Error constraints are nonlinear, hard to converge?
- When/how should elements be added?
- Termination criterion for an optimal solution?

Analysis of the Optimality Conditions

- Error Criteria for Euler-Lagrange Equations
  Requires multiplier estimates (after NLP is solved)

$$S_i(t_{i,nc}) = \begin{bmatrix} \frac{d\lambda^K(t)}{d\tau} + h_i \frac{\partial H^K(t_{i,nc})}{\partial z} \\ \frac{\partial H^K(t_{i,nc})}{\partial y} \\ \frac{\partial H^K(t_{i,nc})}{\partial u} \\ \alpha^{L,K}(t_{i,nc})^T (u^K(t_{i,nc}) - u_L) \\ \alpha^{U,K}(t_{i,nc})^T (u_U - u^K(t_{i,nc}) \end{bmatrix} \qquad \bar{C}\, \|S_i(t_{i,nc})\| \le \varepsilon.$$

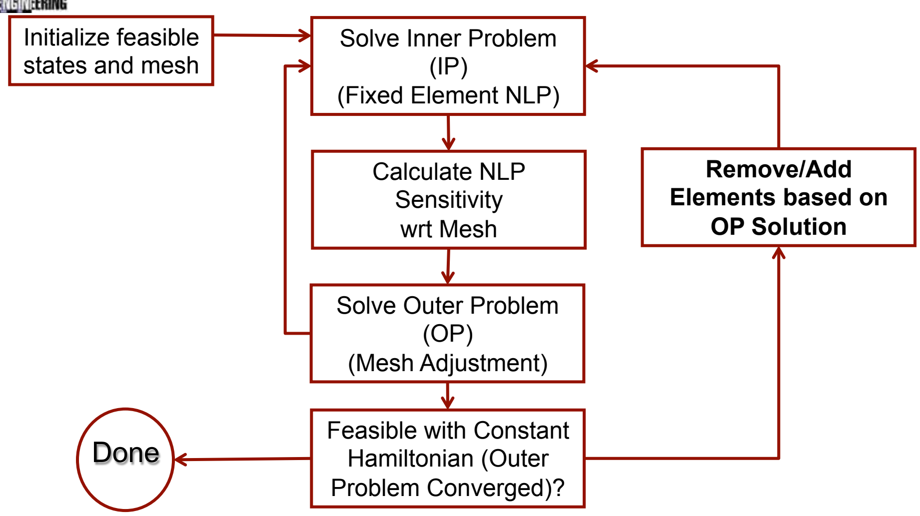
- Alternate Criterion (Constant Hamiltonian Function, Stengel (1994))

$$H_{(i-1)K+k} = H^K(t_{ik}) = \lambda_{ik} f(z_{ik}, y_{ik}, u_{ik}) + \eta_{ik} g(z_{ik}, y_{ik}, u_{ik}) = \frac{\bar{\lambda}_{ik}}{\omega_k h_i} \sum_{j=0}^{K} \dot{\ell}_j(\tau_k) z_{ij}$$

$$\left| H_j - \bar{H} \right| \le \varepsilon_h,\ j = 1, \ldots NK$$

123

Two-Level Optimization Strategy



- Separate direct transcription and mesh adjustment into IP and OP
- Add constant Hamiltonian constraints into OP using adjoints from IP
- Link IP and OP with NLP sensitivity (New Solvers!)

Automated Mesh Initialization

$$
\begin{aligned}
\bar{h}_i \le arg\{\max \quad & h_i \\
s.t. \quad & \sum_{j=0}^{K} \dot{\ell}_j(\tau_k) z_{ij} - h_i f(z_{ik}, y_{ik}, \bar{u}_{ik}) = 0 \\
& z_{i+1,0} = \sum_{j=0}^{K} \ell_j(1) z_{ij} \\
& g(z_{ik}, y_{ik}, \bar{u}_{ik}) = 0, \quad k = 1, K \\
& -\varepsilon \le \bar{C}\, T_i(t_{i,nc}) \le \varepsilon \\
& 0 \le h_i \le min(h_{max}, t_f - \sum_{i'=1}^{i-1} \bar{h}_{i'}),
\end{aligned}
$$

- Choose initial placement of finite elements based on state errors
- Based on initial control profiles *u(t)*
- Feasible for solution of inner problem
- Mimics Implicit Runge-Kutta initial value solver
- “Multiple shooting version” for unstable forward modes

125

Inner Problem (fixed elements)

$$
\begin{aligned}
\min \quad & \Phi(z_f) \\
s.t. \quad & \sum_{j=0}^{K} \dot{\ell}_j(\tau_k) z_{ij} - \bar{h}_i f(z_{ik}, y_{ik}, u_{ik}) = 0 \\
& g(z_{ik}, y_{ik}, u_{ik}) = 0 \\
& u_{ik} = \sigma(\tau_k, v_i),\ u_L \le u_{ik} \le u_U \\
& k \in \{1, \dots, K\},\ i \in \{1, \dots, N\} \\
& z_{i+1,0} = \sum_{j=0}^{K} \ell_j(1) z_{ij}, \quad i = 1, \dots, N-1 \\
& z_f = \sum_{j=0}^{K} \ell_j(1) z_{Nj}, \quad z_{1,0} = z(t_0) \\
& \psi(z_f) \le 0 \\
& T_i = \begin{bmatrix} \frac{dz^K(t)}{d\tau} - h_i f(z^K(t_{i,nc}), y^K(z(t_{i,nc})), u^K(t_{i,nc})) \\ g(z^K(t_{i,nc}), y^K(z(t_{i,nc})), u^K(t_{i,nc})) \end{bmatrix} \\
& i = 1, \dots, N
\end{aligned}
$$

- “Classical” direct transcription formulation
- Embed state error definition with T_i as <u>placeholder sensitivity variables</u>
- Nonsmoothness of KKT conditions for Sensitivity Analysis
- Solve with IPOPT
- Smooth sensitivities from related barrier problem, with $\mu > 0$.

126

Barrier Methods for Large-Scale Nonlinear Programming

Original Formulation

$$\min_{x \in \Re^n} f(x)$$
$$\text{s.t} \quad c(x) = 0$$
$$x \geq 0$$

Can generalize for

$$a \leq x \leq b$$

Barrier Approach

$$\min_{x \in \Re^n} \varphi_\mu(x) = f(x) - \mu \sum_{i=1}^{n} \ln x_i$$
$$\text{s.t} \quad c(x) = 0$$

⇒ **As μ → 0, x*(μ) → x*** **Fiacco and McCormick (1968)**

127

Solution of the Barrier Problem

⇒ **Newton Directions (KKT System)**

$$\begin{aligned} \nabla f(x) + A(x)\lambda - v &= 0 \\ Xv - \mu e &= 0 \\ c(x) &= 0 \end{aligned}$$

$e^T = [1,1,1\ldots],\ X = diag(x)$

$A = \nabla c(x),\ W = \nabla_{xx} L(x,\lambda,v)$

$$\begin{bmatrix} W & A & -I \\ A^T & 0 & 0 \\ V & 0 & X \end{bmatrix} \begin{bmatrix} d_x \\ d_\lambda \\ d_v \end{bmatrix} = - \begin{bmatrix} \nabla f + A\lambda - v \\ c \\ Xv - \mu e \end{bmatrix}$$

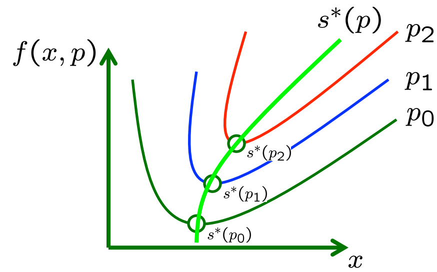
IPOPT Code – www.coin-or.org

128

NLP Sensitivity wrt elements ($\underline{h = p}$)

Parametric Programming

$$\left.\begin{array}{ll} \min & f(x,p) \\ \text{s.t.} & c(x,p) = 0 \\ & x \geq 0 \end{array}\right\} \mathrm{P}(p)$$



Solution Triplet

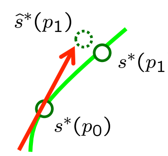
$s^*(p)^T = [x^{*T}\ \lambda^{*T}\ \nu^{*T}]$

Optimality Conditions $\mathrm{P}(p)$

$$\begin{aligned} \nabla_x f(x,p) + \nabla_x c(x,p)\,\lambda - \nu &= 0 \\ c(x,p) &= 0 \\ XVe &= 0 \end{aligned}$$

- **NLP Sensitivity: Rely upon Existence and Differentiability of** $s^*(p)$
- **Main Idea: Obtain** ds^*/dp **and approximate** $s^*(p_1)$ **by Taylor Series Expansion**

$$\hat{s}^*(p_1) \approx s^*(p_0) + \left.\frac{\partial s}{\partial p}^T\right|_{p_0} (p_1 - p_0)$$



129

NLP Sensitivity (sIPOPT)
(Pirnay, Lopez Negrete, B., 2012)

Optimality Conditions of $\mathrm{P}(p)$

$$\left.\begin{aligned} \nabla_x \mathcal{L} = \nabla_x f(x,p) + \nabla_x c(x,p)\,\lambda - \nu &= 0 \\ c(x,p) &= 0 \\ XVe &= 0 \end{aligned}\right\} \mathrm{Q}(s,p) = 0$$

Apply Implicit Function Theorem to $\mathrm{Q}(s,p) = 0$ **around** $(p_0, s^*(p_0))$

$$\frac{\partial \mathrm{Q}(s^*(p_0),p_0)}{\partial s} \left.\frac{\partial s}{\partial p}\right|_{p_0} + \frac{\partial \mathrm{Q}(s^*(p_0),p_0)}{\partial p} = 0$$

$$\begin{bmatrix} W(s^*(p_0)) & A(x^*(p_0)) & -I \\ A(x^*(p_0))^T & 0 & 0 \\ V^*(p_0) & 0 & X^*(p_0) \end{bmatrix} \begin{bmatrix} \frac{\partial x}{\partial p} \\ \frac{\partial \lambda}{\partial p} \\ \frac{\partial \nu}{\partial p} \end{bmatrix} + \begin{bmatrix} \nabla_{x,p}\mathcal{L}(s^*(p_0)) \\ \nabla_p\, c(x^*(p_0)) \\ 0 \end{bmatrix} = 0$$

KKT Matrix from IPOPT

$$\begin{bmatrix} W(x_k,\lambda_k) & A(x_k) & -I \\ A(x_k)^T & 0 & 0 \\ V_k & 0 & X_k \end{bmatrix}$$

- Nonsingular if LICQ and SSOSC hold at $s^*(p_0)$
- Already Factored at Solution
- Sensitivity Calculation from Single Backsolve
- sIPOPT Code embedded within IPOPT

130

Outer Problem (Mesh Adjustment)

$$\min_{\bar{H}, h_i, w_j} \quad \Phi(z_f(h)) + \rho \sum_{j=1}^{NK} w_j$$

$$-\varepsilon \le \bar{C}\, T_i(h) \le \varepsilon$$

$$H_{(i-1)K+k}(h) = \frac{\bar{\lambda}_{ik}(h)}{\omega_k h_i} \sum_{j=}^{K} \dot{\ell}_j(\tau_k) z_{ij}(h)$$

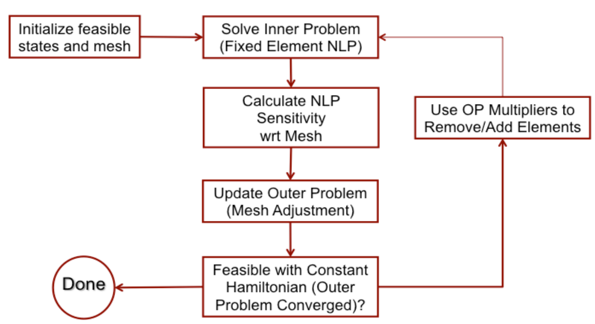
$$-(\varepsilon_h + w_j) \le H_j - \bar{H} \le (\varepsilon_h + w_j),\ w_j \ge 0,\ j = 1, \ldots NK$$

$$0 \le h_i \le h_{max},\ \sum_{i=1}^{N} h_i = t_f,$$

- Few decision variables (2N+1)
- Manipulates element mesh to satisfy state error constraints
- Allows for optimal placement of break points in controls
- Enforces constant Hamiltonian over time
- User-specified tolerances: ε, ε_h
- Generic formulation – fully independent of DAEs
- Sensitivities for $z(h)$, $\lambda(h)$, $T(h)$ obtained from Inner Problem
- Solved with L-BFGS version of IPOPT
- Remove zero elements; bisect and augment maximum elements

131

Demonstration of Two-Level Optimization Strategy

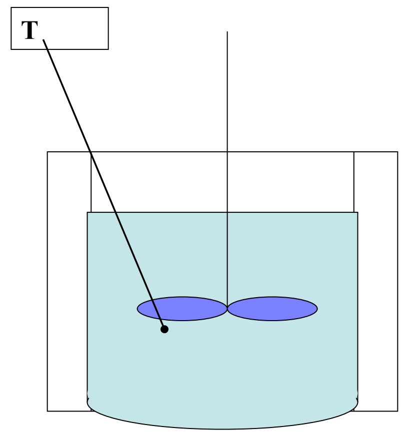


- Strategy implemented in AMPL
- Exact gradient/Hessians for IP, L-BFGS for OP
- Radau Collocation (K=3), Piecewise linear controls
- Solved with IPOPT/sIPOPT; $\varepsilon = \varepsilon_h = 10^{-4}$

- 11 test problems solved (2 – 64 DAEs)
  - Reactor Temperature Profiles
  - Modified Singular Problems

Parallel Batch Reactor
A→B, A→C

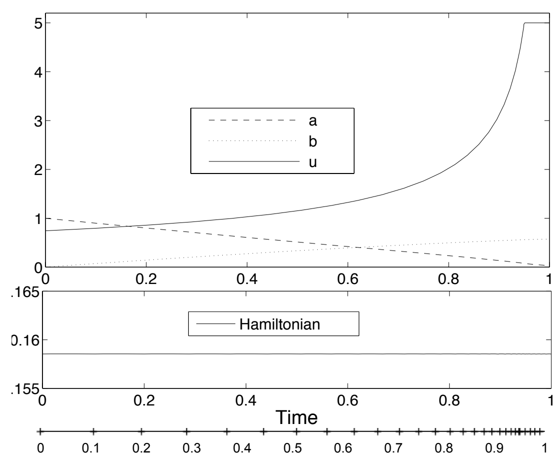
$$
\begin{aligned}
\min \quad & -b(1) \\
s.t. \quad \frac{da}{dt} &= -a(t)\left(u(t)+u(t)^2/2\right) \\
\frac{db}{dt} &= a(t)u(t) \\
a(0) &= 1, \quad b(0)=0, \quad u(t)\in[0,5]
\end{aligned}
$$



Parallel Batch Reactor
A→B, A→C

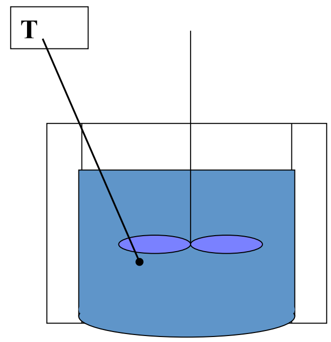
- 20→ 28 elements
- 4 OP Solutions
- 115 IP Iterations
- 43.2 CPUs
- H = 0.1585412

Φ = -0.573544992
- Matches analytic solution

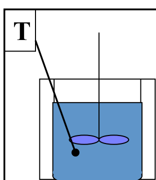


Series Batch Reactor
A→B→C

$$
\begin{aligned}
\min \quad & -b(1) \\
s.t. \quad \frac{da}{dt} &= -a(t)u(t) \\
\frac{db}{dt} &= a(t)\, u(t) - 2\, b(t)\, u(t)^2 \\
a(0) &= 1, \quad b(0) = 0 \\
0 &\leq u(t) \leq 5
\end{aligned}
$$



Series Batch Reactor
A→B→C



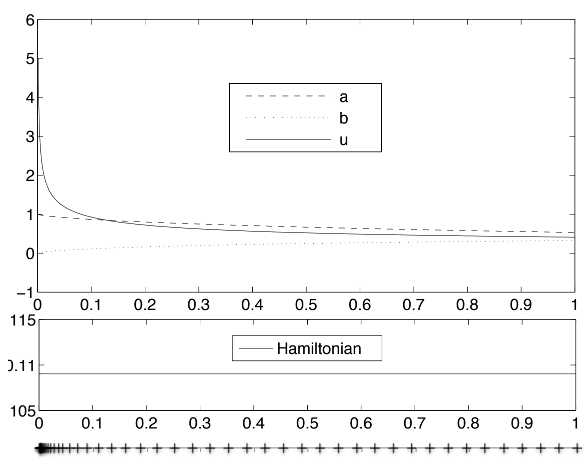
Steep control profile
Sensitive to tolerance

- 20→ 47 elements
- 9 OP Solutions
- 479 IP Iterations
- 439.7 CPUs
- H = 0.1090224625156

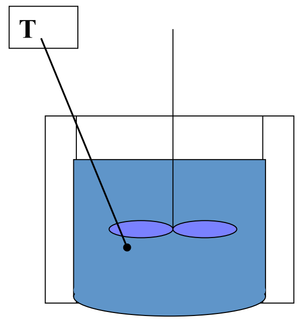
Φ = -0.324143015236

- 30→31 elements
- One OP iteration
- 22.5 CPUs
- H = 0.1090232175342

Φ = -0.324142981371



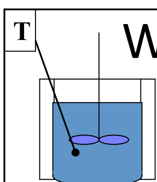
Williams-Otto Batch Reactor (u(t) ≤ 0.6)
$A+B \rightarrow C,\ B+C \rightarrow P+E,\ C+P \rightarrow G$



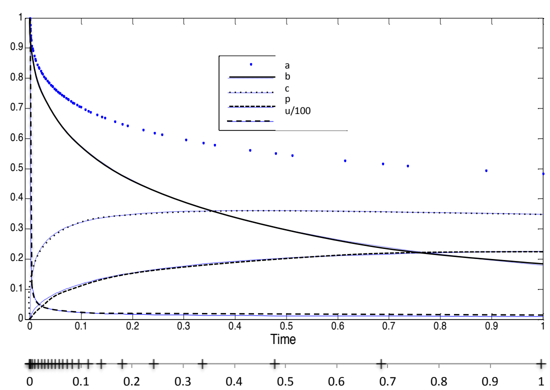
$$
\begin{aligned}
\min \quad & -p(1) \\
s.t. \quad \frac{da}{dt} &= -r_1 \\
\frac{db}{dt} &= -(r_1 + r_2) \\
\frac{dc}{dt} &= 2r_1 - 2r_2 - r_3 \\
\frac{dp}{dt} &= r_2 - r_3 \\
a(0) &= 1, \quad b(0)=1, \quad c(0)=0, \quad p(0)=0, \quad u(t) \in [0,100]
\end{aligned}
$$

$$r_1 = k_1 a(t) b(t),\ r_2 = k_2 b(t) c(t),\ r_3 = k_3 c(t) p(t), \text{ and } k_i = \alpha_i u^{\beta_i}$$

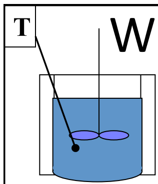
Williams-Otto Batch Reactor (u(t) ≤ 100)
$A+B \rightarrow C,\ B+C \rightarrow P+E,\ C+P \rightarrow G$



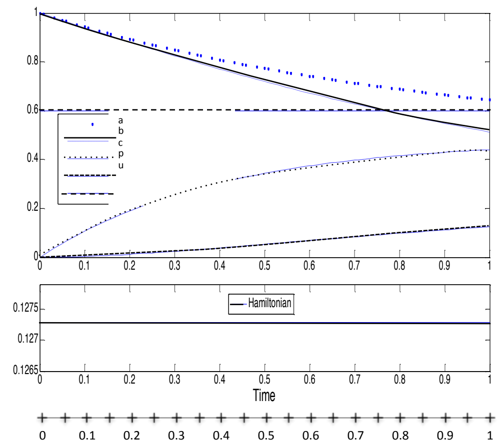
- 20→ 23 elements
- 2 OP Solutions
- Very steep control profile
- First element = 0.0004
- Fails to find constant Hamiltonian



Williams-Otto Batch Reactor (u(t) ≤ 0.6)
A+B→C, B+C→P+E, C+P→G



- 20→ 20 elements
- One OP Solution
- 6 IP Iterations
- 4.6 CPUs
- H = 0.1272836433559
- Φ = -0.1254534249595



Dynamic Optimization of Distillation Column

Maintain Product Purity with Feed Disturbance

s.t. Equilibrium stage model, Binary

Equimolar Overflow (no heat balance)

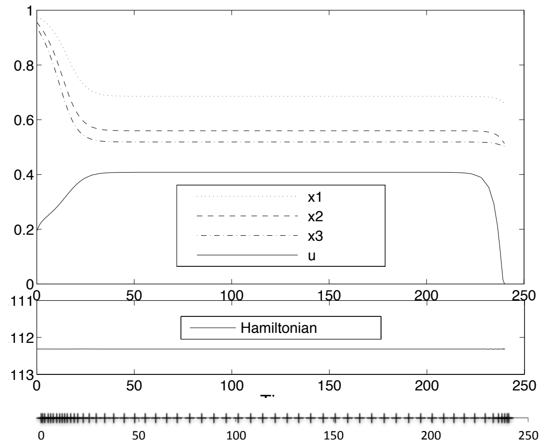
Constant Mass Holdup, adjust distillate rate

32 trays, 64 DAEs

$$
\begin{aligned}
\min \quad & \int_0^{t_f=240} (x_1-0.995)^2 + 0.1u^2 dt \\
s.t. \quad \frac{dx_1}{dt} &= \frac{1}{M_{cond}} V(y_2 - x_1),\ x_1(0) = x_{1,0} \\
\frac{dx_j}{dt} &= \frac{1}{M_{tray}} L(x_{j-1}-x_j) - V(y_j - y_{j+1}),\ x_j(0) = x_{j,0}\ j = 2,\dots 16 \\
\frac{dx_{17}}{dt} &= \frac{1}{M_{tray}} (Fx_f + Lx_{16} - (F+L)x_{17} - V(y_{17}-y_{18}),\ x_{17}(0) = x_{17,0} \\
\frac{dx_j}{dt} &= \frac{1}{M_{tray}} (F+L)(x_{j-1}-x_j) - V(y_j - y_{j+1}),\ x_j(0) = x_{j,0}\ j = 18,\dots 31 \\
\frac{dx_{32}}{dt} &= \frac{1}{M_{reb}} (F+L)x_{31} - (F - u_1(t))x_{32} - Vy_{32},\ x_{32}(0) = x_{32,0} \\
P &= x_j \gamma_{A,j} P_{A,j}^{sat} + (1-x_j)\gamma_{B,j} P_{B,j}^{sat} \\
y_j P &= x_j \gamma_{A,j} P_{A,j}^{sat} \\
P_{A,j}^{sat} &= P_A^{sat}(T_j), \quad P_{B,i}^{sat} = P_B^{sat}(T_j) \\
\gamma_{A,j} &= \gamma_A(x_j, T_j), \quad \gamma_{B,j} = \gamma_B(x_j, T_j)
\end{aligned}
$$

Dynamic Optimization of Distillation Column

- 20→ 56 elements
- 2 OP Solutions
- 64 IP Iterations
- 887.5 CPUs
- H = -0.1123137
- Φ = 24.82098



Implications of KKT Results

- Direct Transcription using Gauss and Radau Collocation - we obtain correct solution for certain classes of problems
- Can use NLP Lagrange multipliers for Adjoint estimation
- Influence of controllability on the convergence analysis – interesting link between control and optimization
- Apply to large-scale, real-world applications if Regular OCPs
- What if Regular OCP assumptions are violated?
  - High index path constraints
  - Singular control problems

  } High-index DAE system

142

High-index DAE Models

$$\left.\begin{aligned}\frac{d}{dt}z_1(t) &= u(t)\\ \frac{d}{dt}z_2(t) &= z_1(t)\\ \frac{d}{dt}z_3(t) &= z_2(t)\end{aligned}\right\}\ \text{Differential Equations}$$

$$\left.z_3(t) = 1\right\}\ \text{Algebraic Equation (does not contain } u(t)\text{)}$$

Differential Equations

Algebraic Equation (does not contain *u(t)*)

143

High-index DAE Models

Differential Equations

$$\boxed{\begin{aligned}\frac{d}{dt}z_1(t) &= u(t)\\ \frac{d}{dt}z_2(t) &= z_1(t)\\ \frac{d}{dt}z_3(t) &= z_2(t)\end{aligned}}$$

Algebraic Equation

$$\boxed{z_3(t) = 1}$$

$$\boxed{\tfrac{d}{dt}z_3(t) = 0}$$

$$\boxed{z_2(t) = 0}$$

$$\boxed{\tfrac{d}{dt}z_2(t) = 0}$$

$$\boxed{z_1(t) = 0}$$

$$\vdots$$

$$\boxed{u(t) = 0}$$

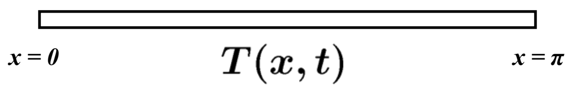
HIGH INDEX

High-index Equations
are hard to solve - numerically

144

Optimal Control of Heat-Conduction Equation

John Betts (Boeing), Stephen Campbell (NCSU)



- Transient Heat Conduction Equation

$$\boxed{\frac{\partial}{\partial t}T = \frac{\partial^2}{\partial x^2}T; \quad 0 \le x \le \pi; \; 0 \le t \le 5}$$

- Initial Condition

$$\boxed{T(x,0) = 0}$$

- We are free to choose the temperature at the boundaries
- Minimize the following objective function

$$\boxed{\int_{t=0}^{5}\int_{x=0}^{\pi} T^2(x,t)\,dx\,dt + 10^{-3}\int_{t=0}^{5}\left(T(0,t)^2(t) + T(\pi,t)^2(t)\right)dt}$$

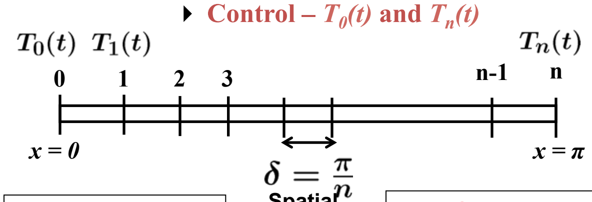
- Path constraint

$$\boxed{T(x,t) \ge \sin(x)\sin(\tfrac{\pi t}{5}) - 0.7}$$

145

Spatial Discretization

- Control – $T_0(t)$ and $T_n(t)$



PDE Optimization Problem — Spatial Discretization → **DAE Optimization Problem**

$$\boxed{\begin{aligned}\frac{\partial}{\partial t}T(x,t) &= \frac{\partial^2}{\partial^2 x}T(x,t)\\ T(x,0) &= 0\end{aligned}}$$

$$\boxed{\begin{aligned}\frac{d}{dt}T_1 &= \frac{T_2 - 2T_1 + T_0}{\delta^2}; \quad T_1(0) = 0;\\ \frac{d}{dt}T_2 &= \frac{T_3 - 2T_2 + T_1}{\delta^2}; \quad T_2(0) = 0;\\ &\vdots\\ \frac{d}{dt}T_{n-1} &= \frac{T_n - 2T_{n-1} + T_{n-2}}{\delta^2}; \quad T_{n-1}(0) = 0;\end{aligned}}$$

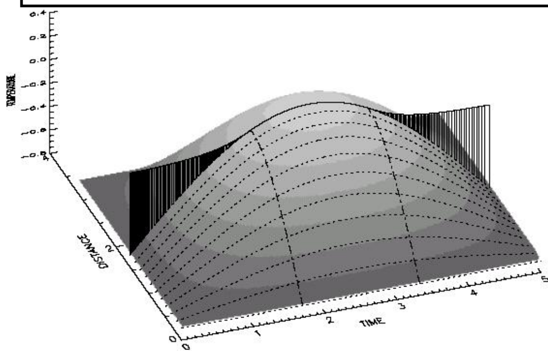
$$\boxed{T_k(t) \;\ge\; sin(k\delta)sin\left(\frac{\pi t}{5}\right) - 0.7; \quad k = 0, \ldots, n}$$

146

Path Constraints

- Computational experience – Impose constraint only at the center of the rod

$$\boxed{T_{\frac{n}{2}}(t) \;\geq\; sin\left(\frac{n}{2}\delta\right) sin\left(\frac{\pi t}{5}\right) - 0.7}$$



- Optimal solution – temperature rides ("active") on the constraint (center)
- Control – boundary temperature

147

Symmetry in DAE Optimization Problem

- n – even number

$$T_k(t) = T_{n-k}(t); \quad k = 0, \ldots, \frac{n}{2} - 1$$

- Time scaling; $t = \delta^2 \tau$

$$\min_{T_0(\tau)} \; \frac{1}{2}\left(\int_0^{5\delta^{-2}} 4\delta^3 \sum_{k=1}^{\frac{n}{2}-1} T_k^2(\tau)d\tau + \int_0^{5\delta^{-2}} 2\delta^3 T_{\frac{n}{2}}^2(\tau)d\tau\right) + \frac{1}{2}\left(2\delta^2(\delta+2q)\int_0^{5\delta^{-2}} T_0^2(\tau)d\tau\right)$$

$$\begin{aligned}
\frac{dT_1}{d\tau} &= T_2 - 2T_1 + T_0; \quad T_1(0) = 0;\\
\frac{dT_2}{d\tau} &= T_3 - 2T_2 + T_1; \quad T_2(0) = 0;\\
&\vdots\\
\frac{dT_{\frac{n}{2}-1}}{d\tau} &= T_{\frac{n}{2}} - 2T_{\frac{n}{2}-1} + T_{\frac{n}{2}-2}; \quad T_{\frac{n}{2}-1}(0) = 0;\\
\frac{dT_{\frac{n}{2}}}{d\tau} &= 2T_{\frac{n}{2}-1} - 2T_{\frac{n}{2}}; \quad T_{\frac{n}{2}}(0) = 0;\\
T_{\frac{n}{2}}(\tau) &\geq sin(\frac{n}{2}\delta)sin(\frac{\pi\delta^2\tau}{5}) - 0.7
\end{aligned}$$

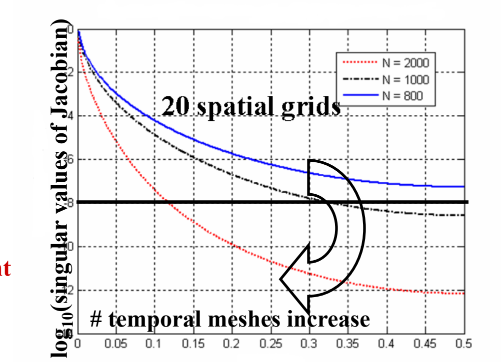
- High-Index Problem – Index = n/2 + 1
- Very hard to solve numerically

148

Direct Simultaneous Approach

PDE Optimization Problem → Spatial Discretization → **DAE Optimization Problem** → Temporal Discretization → **NLP**

- Betts and Campbell (2003) – "Direct Method Works! But it shouldn't"
- Fraction of independent active constraints decreases with an increase in number of temporal meshes.
- Active set SQP discards a large fraction of active constraints.
- **Linear independence of constraint gradients fails to hold (to a given tolerance level)**



Solution Profiles – Betts-Campbell Problem

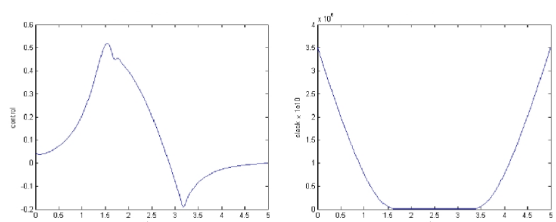


Figure 1: Plots showing the control profile and the slack profile (scaled by 10^{10}) corresponding to the inequality constraint for $y_{\frac{n}{2}}$ (see (15)), obtained using an explicit Euler discretization, for $n = 10$ and $N = 2000$.

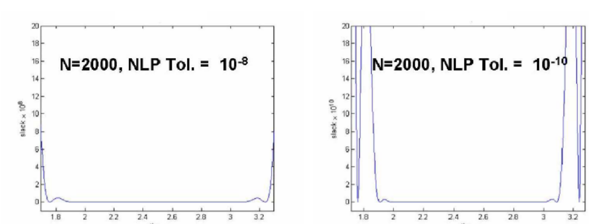


Figure 6: Plots showing the slack profiles (scaled by 1/NLP Tol.), over the potentially active region, for $N = 2000$ and for different NLP solver tolerances, for $n = 10$.

Inequality Multiplier Profiles
Betts-Campbell Problem

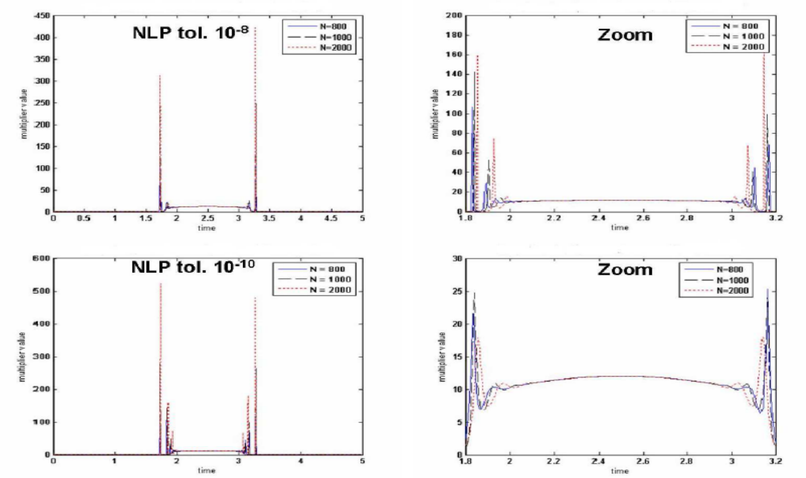


Figure 9: Plots comparing the multiplier profile ($\mu^*[i]$ vs $\frac{5}{N} \times i$), for $n = 20$, for various values of N. The effect of NLP solver tolerance is also shown.

151

NLP Based Methods Could Help !

- Mangasarian-Fromowitz Constraint Qualification (MFCQ) is satisfied (bounded but nonunique multipliers)
- Adjoint Estimation – MFCQ means that NLP multipliers are bounded but not unique (IPOPT Convergence - Analytic Center of the set of multipliers)
- Barrier methods retain all constraints and have convergence results with MFCQ
- For even index ≥ 4, a path constraint cannot be active over nonzero interval, only "touch and goes" are possible (Jacobsen et.al. 1971)
- Variational (Indirect) approach based on IV-solvers may not work – stability and error accumulation
- NLP based methods – flexibility w.r.t CQs – can be used to obtain meaningful solutions for state profiles (but not adjoint profiles)

152

Singular Optimal Control Problems

Singular optimal control problem

$$\min_{u(t)} \quad \varphi\left(z(t_f)\right)$$

$$\text{s.t.} \quad \frac{d}{dt}z = f(z) + ug(z); \qquad z(0) = z_0$$

$$a \leq u(t) \leq b$$

- **Control variable appears <u>linearly</u> in the governing equations and the objective function**
- **Applications**
  - **- Optimal trajectories for space maneuvers – Thrust – control variable**
  - **- Process Engineering – Flow rate – control variable**
- **Scalar control variable – can be extended to vector controls**

153

Singular Optimal Control Problems

Singular optimal control problem

$$\min_{u(t)} \quad \varphi\left(z(t_f)\right)$$

$$\text{s.t.} \quad \frac{d}{dt}z = f(z) + ug(z); \qquad z(0) = z_0$$

$$a \leq u(t) \leq b$$

Hamiltonian :

$$H = \lambda^T f(z) + u\lambda^T g(z) + \mu_1(a - u) + \mu_2(u - b)$$

- **Necessary condition of optimality – Pontryagin's principle**
- **Strictly between bounds, necessary conditions alone fail to determine *u(t)***

$$\boxed{H_u = \lambda^T g(z) = 0}$$

Not a function of *u*

- **Repeated differentiation is required to determine control – High index (≥ 3) constraint for indirect approach – CUMBERSOME for large problems**

154

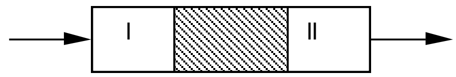
Example: Catalyst Mixing Problem

$$\min \quad a(t_f) + b(t_f) - a_0$$

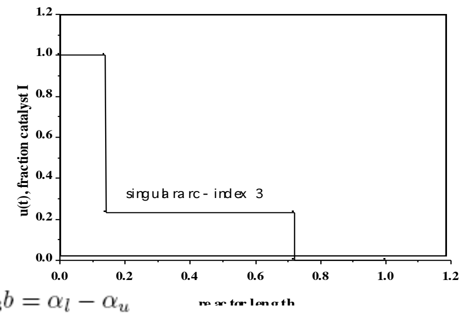
$$s.t. \quad \frac{da}{dt} = -u(k_1 a - k_2 b)$$

$$\frac{db}{dt} = u(k_1 a - k_2 b) - (1-u)k_3 b$$

$$a(0) = a_0, \quad b(0) = 0$$



A <==> B --> C
I II



$$\frac{d\lambda_1}{dt} = -(\lambda_2 - \lambda_1)k_1 u$$

$$\frac{d\lambda_2}{dt} = (\lambda_2 - \lambda_1)k_2 u + (1-u)\lambda_2 k_3$$

$$\lambda_1(t_f) = 1, \quad \lambda_2(t_f) = 1$$

$$\frac{\partial H}{\partial u} = J = (\lambda_2 - \lambda_1)(k_1 a - k_2 b) + \lambda_2 k_3 b = \alpha_l - \alpha_u$$

$$0 \le \alpha_l(t) \perp u(t) \ge 0$$

$$0 \le \alpha_u(t) \perp (1 - u(t)) \ge 0$$

155

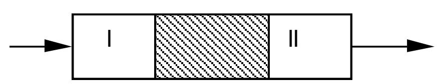
Example: Catalyst Mixing Problem

$$\min \quad a(t_f) + b(t_f) - a_0$$

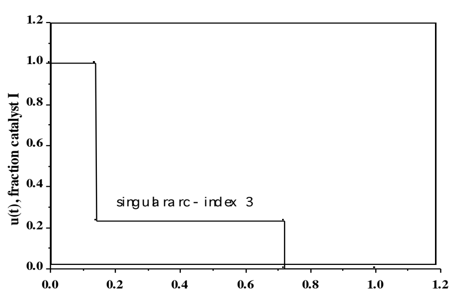
$$s.t. \quad \frac{da}{dt} = -u(k_1 a - k_2 b)$$

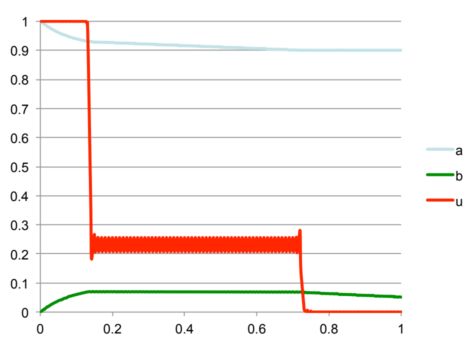
$$\frac{db}{dt} = u(k_1 a - k_2 b) - (1-u)k_3 b$$

$$a(0) = a_0, \quad b(0) = 0$$



A <==> B --> C
I II

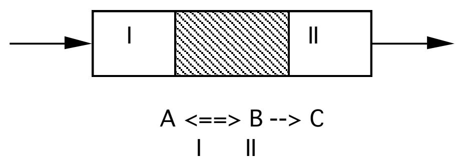


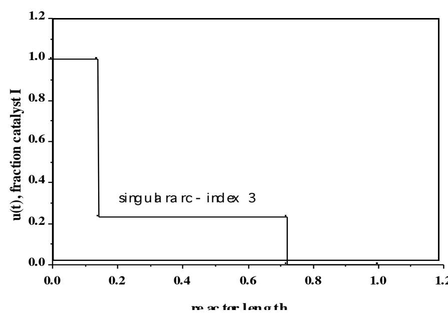


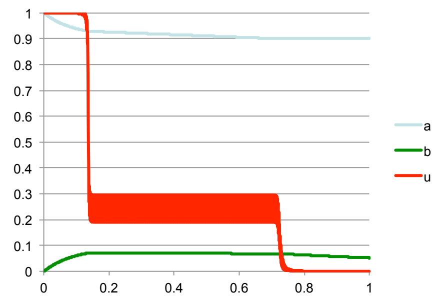
K = 3, N = 100

Example: Catalyst Mixing Problem

$$
\begin{aligned}
\min \quad & a(t_f) + b(t_f) - a_0 \\
s.t. \quad & \frac{da}{dt} = -u(k_1 a - k_2 b) \\
& \frac{db}{dt} = u(k_1 a - k_2 b) - (1-u)k_3 b \\
& a(0) = a_0, \quad b(0) = 0
\end{aligned}
$$



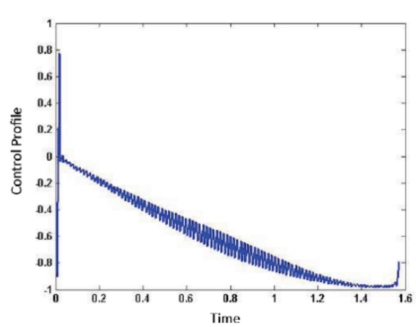




K = 3, N = 500

Direct Simultaneous Approach - Singular OCPs

- While proving convergence results - $\left\| \left(Z^T H Z\right)^{-1} \right\|_\infty = O\left(\frac{1}{h}\right)$
- Singular Problems - Ill-conditioning - $\left\| \left(Z^T H Z\right)^{-1} \right\|_\infty = O\left(\frac{1}{h^q}\right); \quad q > 2$
- Similar problems occur with all other approaches



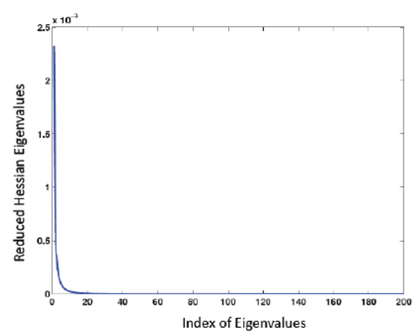


Figure 10.10. *Control profile obtained for $N = 100$, $K = 2$ for problem (10.50).*

Direct Simultaneous Approach - Singular OCPs

- While proving convergence results - $\left\| \left(Z^T H Z \right)^{-1} \right\|_\infty = O\left(\frac{1}{h}\right)$
- Singular Problems - Ill-conditioning - $\left\| \left(Z^T H Z \right)^{-1} \right\|_\infty = O\left(\frac{1}{h^q}\right); \quad q > 2$
- Similar problems occur with all other approaches

Proposed Approach

- Based on Indirect Approach - discretize optimality conditions
- Calculating higher derivatives of H_u –cumbersome for large problems
- Ill-conditioning (not enough info) – no differentiation) "REGULARIZATION"
- Regularization - providing more information to the ill-conditioned problem
  - Use Radau collocation – well suited for this problem
  - Ensure that the H_u profile is smooth
  - Monitor error in the ODE residuals

159

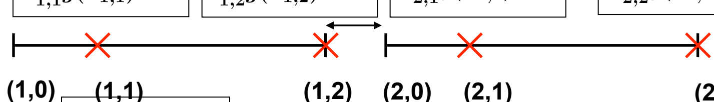
How Continuous is H_u ?

- Totally singular problems – necessary conditions solved using Radau collocation
- Radau – well suited for DAEs (and high-index), highest precision after Gauss, good stability properties – also clear from reduced Hessian results

$$\min_{x,u} \quad \phi\left(x(t_f)\right)$$
$$\text{s.t.} \quad \frac{dx}{dt} = f(x) + ug(x); \qquad x(0) = x_0$$

Hamiltonian :
$$H = \lambda^T f(x) + u(\lambda^T g(x))$$
$$H_u = \lambda^T g(x) = 0$$

$\lambda_{1,1}^T g(x_{1,1}) = 0$ $\quad$ $\lambda_{1,2}^T g(x_{1,2}) = 0$ $\quad$ $\lambda_{2,1}^T g(x_{2,1}) = 0$ $\quad$ $\lambda_{2,2}^T g(x_{2,2}) = 0$



$$\lambda_{1,2} = \lambda_{2,0}$$
$$x_{1,2} = x_{2,0}$$

Continuity of states and adjoints across elements

$$\lambda_{1,2}^T g(x_{1,2}) = \lambda_{2,0}^T g(x_{2,0}) = 0$$

160

How Continuous is H_u ?

- Totally singular problems – necessary conditions solved using Radau collocation
- Radau – well suited for DAEs (and high-index), highest precision after Gauss, good stability properties – also clear from reduced Hessian results

$$\min_{x,u} \quad \phi\left(x(t_f)\right)$$
$$\text{s.t.} \quad \frac{dx}{dt} = f(x) + ug(x); \qquad x(0) = x_0$$

Hamiltonian :

$$H = \lambda^T f(x) + u(\lambda^T g(x))$$
$$H_u = \lambda^T g(x) = 0$$

$$\lambda_{i,j}^T g(x_{i,j}) = 0; \quad i = 1, \ldots, NE; \quad j = 1, \ldots, K$$
$$\Downarrow$$
$$\lambda_{i,0}^T g(x_{i,0}) = 0; \quad i = 2, \ldots, NE;$$

Advantage of using Radau collocation

$\lambda_{1,0}^T g(x_{1,0})$ should be made as close to zero as possible

Additional Information

H_u at the start of the singular segment has to be forced to zero

need higher time derivatives of H_u to be zero.

Practical Considerations at this Point

- Ill-conditioned system – Absolutely essential to set tight solver tolerances
- May not want to drive step-size very small

TSP-1: Aly and Chan (1973)
Totally Singular problem
$u^* = -sin(t)$

$$\min_u \quad x_3\left(\frac{\pi}{2}\right)$$
$$\text{s.t.} \quad \frac{d}{dt}x_1 = x_2; \quad x_1(0) = 0$$
$$\frac{d}{dt}x_2 = u; \quad x_2(0) = 1$$
$$\frac{d}{dt}x_3 = \frac{1}{2}x_2^2 - \frac{1}{2}x_1^2; \quad x_3(0) = 0$$
$$-1 \le u \le 1$$

Practical Considerations at this Point

- Ill-conditioned system – Essential to set tight solver tolerances
- May not want to drive step-size very small

$$\boxed{\begin{array}{l}\dfrac{d}{dt}\lambda_2 = -\lambda_1 - x_2; \quad \lambda_2(t_f) = 0 \\ H_u = \lambda_2(t) = 0\end{array}} \longrightarrow \boxed{\begin{array}{l}C\Lambda_2 + h\,(\Lambda_1 + X_2) = 0 \\ \Lambda_2 = 0\end{array}}$$

$$\downarrow \qquad\qquad\qquad\qquad \downarrow$$

$$\boxed{\lambda_1(0) + x_2(0) = 0} \qquad \boxed{\|\Lambda_1 + X_2\|_\infty = O\left(\frac{1}{h}\right)}$$

- Consistent initial conditions ?
- Use of *h* as a regularization parameter - *a posteriori* determination of regularization parameter can be employed

163

Is this Sufficient ?

- ODE Residual – responsible for spikes
- Minimizing residual error - **KEY**

$$r_i^x(\tau_j) = \sum_{k=0}^{K} \dot{\ell}_k(\tau_j) x_{ik} - h_i f(x_{ij}, u_{ij}) = 0; \quad i = 1, NE\ j = 1, K$$

$$r_i^x(0) = \frac{1}{h_i} \sum_{k=0}^{K} \dot{\ell}_k(0) x_{ik} - f(x_{i-1,K}, \hat{u}_{i0}); \quad i = 1, NE$$

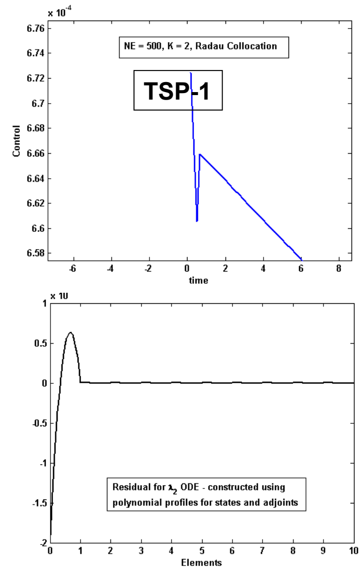
$$r_i(0) = \begin{bmatrix} r_i^x(0) \\ r_i^\lambda(0) \end{bmatrix}; \quad i = 1, \ldots, NE$$

TRY TO MINIMIZE $r_i(0)\ \forall i$

164

Basis for Minimizing the Residuals

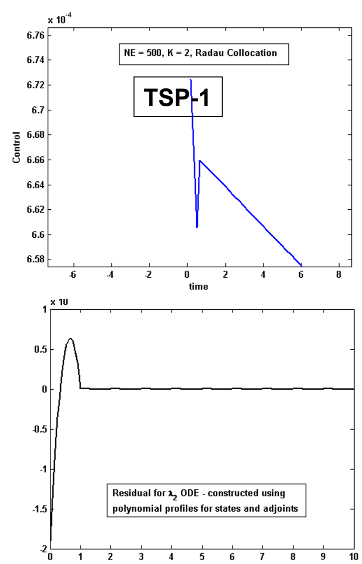
- ❑ Numerical Observations
- ❑ Oscillations due to build up of error. By minimizing residual – error is reduced



165

Basis for Minimizing the Residuals

- ❑ Numerical Observations
- ❑ Oscillations due to build up of error. By minimizing residual – error is reduced



166

Basis for Minimizing the Residuals

- Numerical Observations
- Oscillations due to build up of error. By minimizing residual – error is reduced
- Adds extra smoothness to the states and the adjoints

TSP-1

$$r_i(\tau) = \sum_{k=0}^{K} \dot{\ell}_k(\tau) x_{ik} - h_i f(x_i(\tau), u_i(\tau)); \ \forall i$$

↓

Polynomial of degree K-1

By minimizing the residual at the start of an elements, we try to make a polynomial of degree K-1 behave like a polynomial of degree K

$$r_i^x(0) = \sum_{k=0}^{K} \dot{\ell}_k(0) x_{ik} - h_i f(x_{i-1,K}, \hat{u}_{i0}); \ \forall i$$

167

Basis for Minimizing the Residuals

- Numerical Observations
- Oscillations due to build up of error. By minimizing residual – error is reduced
- Adds extra smoothness to the states and the adjoints

$$\frac{d}{dt}\lambda_2 = -\lambda_1 - x_2; \quad \lambda_2(t_f) = 0$$
$$H_u = \lambda_2(t) = 0$$

- Attempt a consistent initial condition

$$\lambda_2(t_0) = \lambda_2(t_0 + h\tau_1) = \lambda_2(t_0 + h\tau_2) = 0$$

- Error at non-collocation point
  - – strategy for elemental placement
  - – ensure accurate differential variable profiles
  - – also used in the partially singular case

$$r_i^{\lambda_2}(0) = -h_1\left(\lambda_1(0) + x_2(0)\right)$$

168

Totally Singular Problem - Regularization

$$\min_{u(t)} \quad \varphi(z(t_f))$$
$$\text{s.t.} \quad \frac{d}{dt}z = f(z) + ug(z); \quad z(0) = z_0$$

Discretized State Equations
+
Discretized Adjoint Equations
+
Boundary conditions
+
Discretized $H_u = 0$ constraint

- **Indirect approach without index reformulation – ill-conditioned problem**

$$\min \sum_{i=1}^{NE} \delta_i + \alpha \sum_{i=1}^{NE} h_i \varepsilon_i$$

s.t.

Discretized State Equations
+
Discretized Adjoint Equations
+
Boundary conditions
+
Discretized $H_u = 0$ constraint
+

$$-\varepsilon_i e_r \le \begin{bmatrix} r_i^z(0) \\ r_i^\lambda(0) \end{bmatrix} \le \varepsilon_i e_r; \; i = 1, \ldots, NE$$

$$-\delta_i \le \lambda_{i,0}^T g(z_{i,0}) \le \delta_i; \; i = 1, \ldots, NE$$

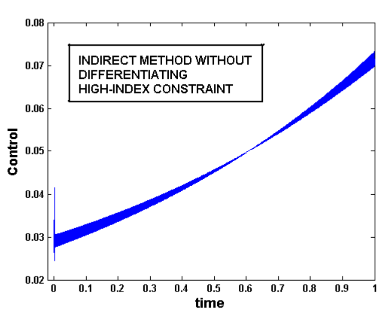
169

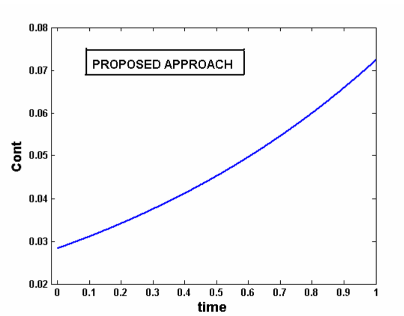
Optimal Control of a Bioreactor

TSP: Menawat et.al. (1987)

Semicontinous fermentor with biomass as the primary product Maximize production of biomass

$$\min_u \quad -x(1)V(1)$$
$$\text{s.t} \quad \frac{d}{dt}x = \frac{sx}{(10^{-3}+s)(1+s)} - \frac{xu}{V}; \; x(0) = 0.3$$
$$\frac{d}{dt}s = \frac{u}{V}(10 - s) - \frac{sx}{(10^{-3}+s)(1+s)}; \; s(0) = \sqrt{10^{-3}}$$
$$\frac{d}{dt}V = u; \; V(0) = 1$$
$$0 \le u \le 10^{-1}$$





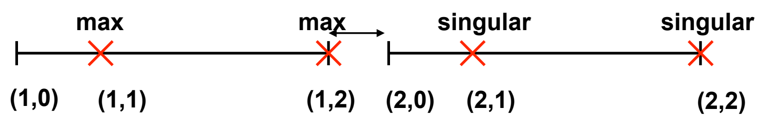
170

Partially Singular Problems (e.g., Bang-bang and singular)
What should the formulation consider ?

- **Switching**

- **Prevent switching within an element – controls at all collocation points within an element should all be at a bound or strictly between bounds**

- **"bang-singular" or "singular-bang" – the multiplier on the constrained side has a non-strict complementary relation with the control variable – formulation should not prevent this**



$$\boxed{\mu_{1,2}^{(u)}(u_{max} - u_{1,2}) = 0} \xrightarrow{\text{non-strict}} \boxed{\mu_{1,2}^{(u)} = 0 \ \& \ u_{1,2} = u_{max}}$$

- **Objective function described in the previous section – valid only over singular segment**

COMPLEMENTARITY (EQUILIBRIUM CONSTRAINTS) BASED FORMULATION

171

Complementarity Constraints

- **Way of modeling certain discrete decisions – Switching**

$$\boxed{\mathrm{x}^T\mathrm{y} = 0; \quad \mathrm{x}, \mathrm{y} \geq 0; \quad \mathrm{x}, \mathrm{y} \in \Re^n} \quad \boxed{x \geq 0 \perp y \geq 0}$$

- **Violates standard regularity assumptions (LICQ and MFCQ) associated with NLPs – lack of an interior**

- **Add constraints x,y ¸ 0, but add the term $x^T y$ to the objective function with a large weight – ℓ1 penalty (Leyffer and Nocedal, 2003)**

$$\boxed{\mathrm{x}^T\mathrm{y} \leq c\mu; \quad \mathrm{x}, \mathrm{y} \geq 0; \quad \mathrm{x}, \mathrm{y} \in \Re^n}$$

μ barrier parameter, driven down to zero

- **Meshes well with interior point algorithms (IPOPT)**

- **Convergence results (Raghunathan, B., 2002).**

172

Complementarity Formulation

- Instead of forcing complementarity at each collocation point – force complementarity relations with maximum deviations from the bounds

$$\max_j \left\{\mu_{i,j}^{(u)}\right\} \geq 0 \perp \max_j \left\{u_{max} - u_{i,j}\right\} \geq 0$$
$$\max_j \left\{\mu_{i,j}^{(\ell)}\right\} \geq 0 \perp \max_j \left\{u_{i,j} - u_{min}\right\} \geq 0$$

$$0 \leq \mu_{i,j}^{(u)} \leq \epsilon_i^{\mu^{(u)}} \leq M_1; \quad \forall j$$
$$0 \leq u_{max} - u_{i,j} \leq \epsilon_i^{umax} \leq M_2; \quad \forall j$$

$$\epsilon_i^{\mu^{(u)}} \geq 0 \perp \epsilon_i^{umax} \geq 0$$

control = $\{MAX, SING, MIN\}$

If $u_{i,1}$ and $u_{i,2}$ belong to different arcs e.g. MAX - SING, MAX - MIN then all the bound multipliers for that element (i) are zero

173

Optimization Problem

- Complementarity formulation
  - Make step-size h_i also a variable
  - Use capability of residual error term to place elements

$$(1 - \tfrac{f}{100})\tfrac{t_f - t_0}{NE} \leq h_i \leq (1 + \tfrac{f}{100})\tfrac{t_f - t_0}{NE}$$
$$\sum_{i=1}^{NE} h_i = t_f - t_0$$

- Objective for TSP can be used – valid only over singular arc – else affect multipliers too much
- Use multipliers to decide - singular or not
- Can also use complementarity – need to test

$$\alpha_2 = O\left(\frac{\alpha_1}{10\hat{\epsilon}}\right)$$
$$\hat{\epsilon} = 10^{-3} \sim 10^{-4}$$

$$\min \quad \alpha_1 \sum_{i=1}^{NE} \left(\frac{\delta_i}{\hat{\epsilon} + \sum_{j=1}^{2} (\mu_{i,j}^{(\ell)} + \mu_{i,j}^{(u)})} \right) + \alpha_2 \sum_{i=1}^{NE} h_i \varepsilon_i$$

s.t.

Discretized State and Adjoint Equations
+
Boundary conditions
+
Discretized $H_u = 0$ constraint
+
Complementarity conditions
+
Element length constraints
+

$$-\varepsilon_i e_r \leq \begin{bmatrix} r_i^x(0) \\ r_i^\lambda(0) \end{bmatrix} \leq \varepsilon_i e_r; \; i = 1, \ldots, NE$$

$$-\delta_i \leq \lambda_{i,0}^T g(x_{i,0}) - \hat{\mu}_{i,0}^{(\ell)} + \hat{\mu}_{i,0}^{(u)} \leq \delta_i; \; i = 1, \ldots, NE$$

174

Example Problems – Partially Singular OCPs

PSP-1: Aly (1978)
Min arc - Singular arc

$$\begin{aligned}\min_u \quad & x_3(5)\\ \text{s.t.} \quad & \frac{d}{dt}x_1 = x_2; \quad x_1(0)=0\\ & \frac{d}{dt}x_2 = u; \quad x_2(0)=1\\ & \frac{d}{dt}x_3 = \frac{1}{2}x_2^2 + \frac{1}{2}x_1^2; \quad x_3(0)=0\\ & -1 \le u \le 1\end{aligned}$$

PSP-2: Jackson (1968)
Catalyst mixing problem
Max - Singular - Min

$A \rightleftharpoons B \rightarrow C$

catalyst 1 | catalyst 2

Maximize mole fraction of C

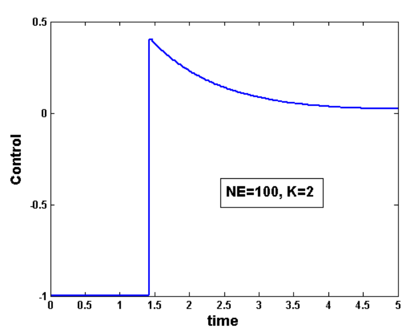
$$\begin{aligned}\min_u \quad & x_1(1) + x_2(1)\\ \text{s.t.} \quad & \frac{dx_1}{dt} = u(k_2x_2 - k_1x_1); \quad x_1(0)=1\\ & \frac{dx_2}{dt} = u(k_1x_1 - k_2x_2) - k_3(1-u)x_2; \quad x_2(0)=0\\ & 0 \le u \le 1\end{aligned}$$

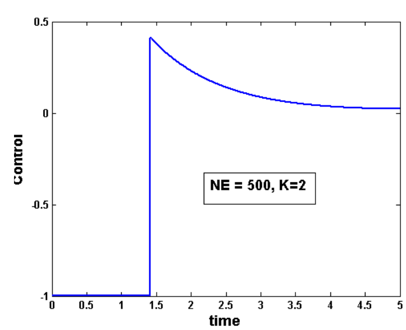
175

Results – Test Example 4

PSP-1: Aly (1978)
Min arc - Singular arc

$$\begin{aligned}\min_u \quad & x_3(5)\\ \text{s.t.} \quad & \frac{d}{dt}x_1 = x_2; \quad x_1(0)=0\\ & \frac{d}{dt}x_2 = u; \quad x_2(0)=1\\ & \frac{d}{dt}x_3 = \frac{1}{2}x_2^2 + \frac{1}{2}x_1^2; \quad x_3(0)=0\\ & -1 \le u \le 1\end{aligned}$$



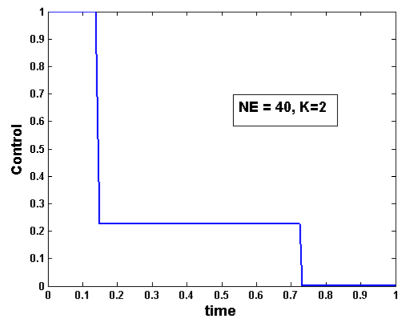


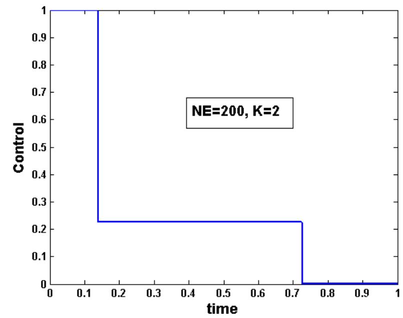
176

Results – Test Example 5

PSP-2: Jackson (1968)
Catalyst mixing problem
Max - Singular - Min

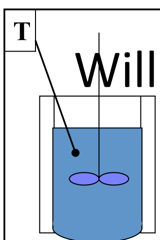
$$\min_u \quad x_1(1) + x_2(1)$$
$$\text{s.t.} \quad \frac{dx_1}{dt} = u(k_2 x_2 - k_1 x_1); \qquad x_1(0) = 1$$
$$\frac{dx_2}{dt} = u(k_1 x_1 - k_2 x_2) - k_3(1 - u)x_2; \qquad x_2(0) = 0$$
$$0 \le u \le 1$$





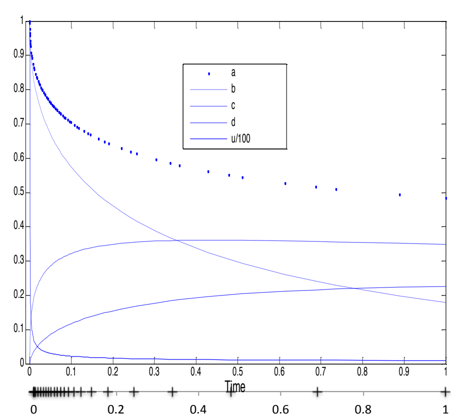
- Predicts switching times and value over singular arc accurately – no switching within element – permits non-complementary multipliers
- Need to investigate formulations that converge faster

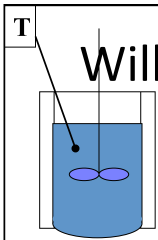
177



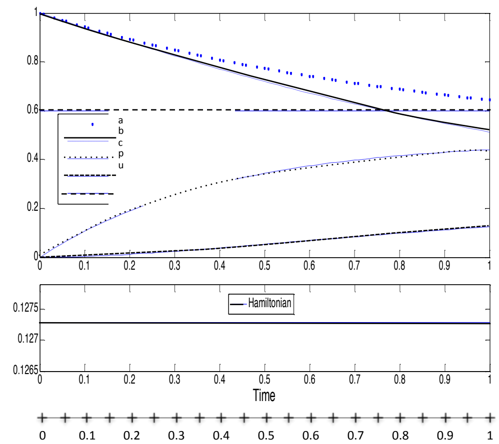
Williams-Otto Batch Reactor (u(t) ≤ 100)

A+B→C, B+C→P+E, C+P→G





Williams-Otto Batch Reactor (u(t) ≤ 0.6)
$A+B \rightarrow C$, $B+C \rightarrow P+E$, $C+P \rightarrow G$



20→ 20 elements
One OP solution
11 IPOPT Inner Iterations
2.5 CPUs
H = 0.02382238
Φ = -0.06573002542291

Catalyst Mixing Singular Problem

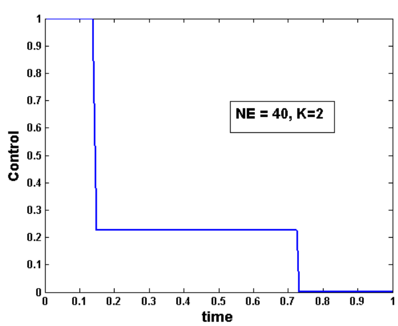
PSP-2: Jackson (1968)
Catalyst mixing problem
Max - Singular - Min

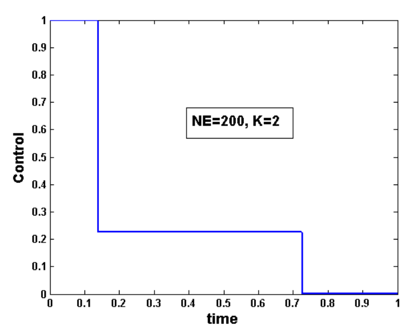
$$\min_{u} \quad x_1(1) + x_2(1)$$

$$\text{s.t.} \quad \frac{dx_1}{dt} = u(k_2 x_2 - k_1 x_1); \qquad x_1(0) = 1$$

$$\frac{dx_2}{dt} = u(k_1 x_1 - k_2 x_2) - k_3(1 - u)x_2; \qquad x_2(0) = 0$$

$$0 \leq u \leq 1$$



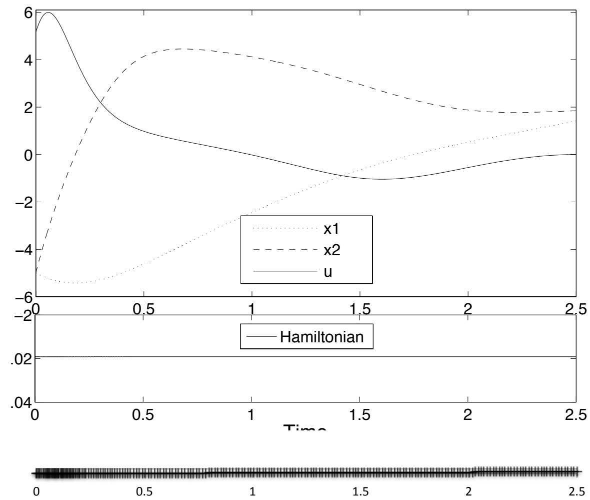


- ❑ **Predicts switching times and value over singular arc accurately – no switching within element – permits non-complementary multipliers**
- ❑ **Need to investigate formulations that converge faster**

180

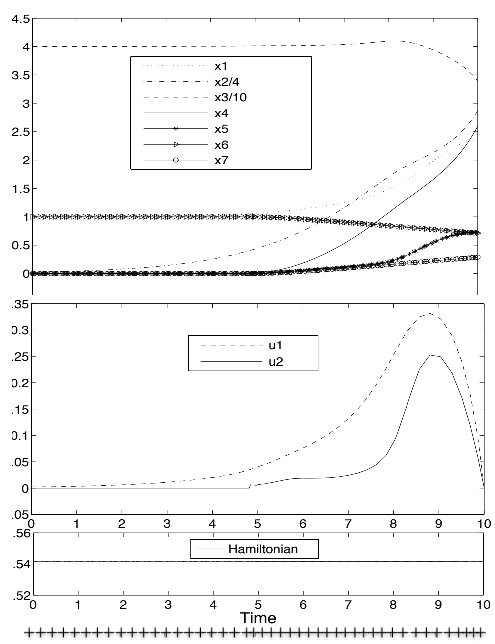
Rayleigh Problem

- 20→ 208 elements
- 4 OP Solutions
- 428 IP Iterations
- 1788.4 CPUs
- H = -2.019174
- Φ = 29.37608



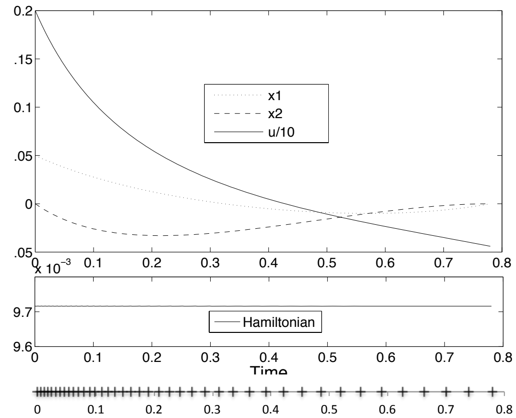
Lee-Ramirez Bioreactor

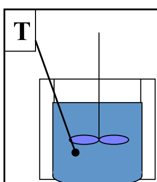
- 20→ 44 elements
- 2 OP Solutions
- 39 IPOPT Inner Iterations
- 1241.38 CPUs
- H = 3.54156591
- Φ = -6.0995216123



Kirk Stirred Tank Reactor
Optimal Feeding Profile

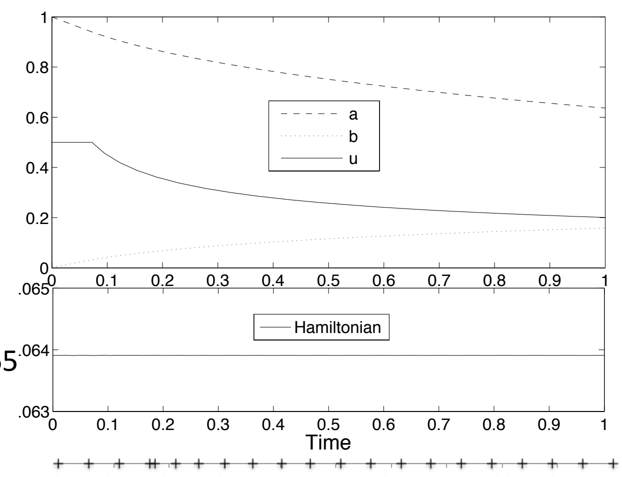
- 20→ 40 elements
- 2 OP Solutions
- 39 IPOPT Inner Iterations
- 16.1 CPUs
- H = 0.009716093
- Φ = 0.0167028





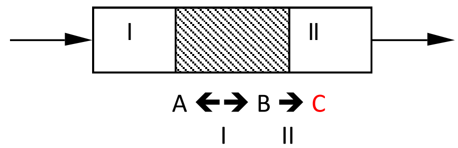
Van de Vusse Batch Reactor
A→B→C, 2A→D

- 20→ 21 elements
- 2 OP Solutions
- 42 IP Iterations
- 11.7 CPUs
- H = 0.06390922
- Φ = -0.1585018546365



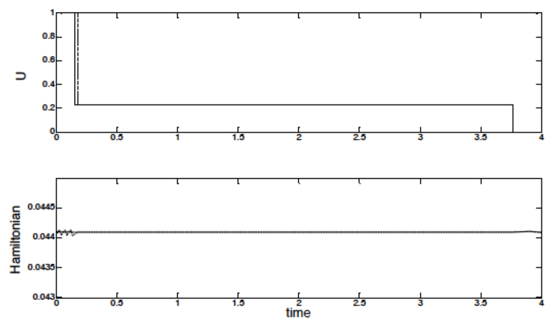
Modified Catalyst Mixing Problem

$$
\begin{aligned}
\min \quad & a(t_f) + b(t_f) - a_0 + 0.1\int_0^{t_f} u^2 dt \\
s.t. \quad \frac{da(t)}{dt} &= -u(k_1 a(t) - k_2 b(t)) \\
\frac{db(t)}{dt} &= u(k_1 a(t) - k_2 b(t)) - (1-u)k_3 b(t) \\
a(0) &= 1, \quad b(0) = 0, \quad u(t) \in [0,1].
\end{aligned}
$$



Solved with earlier MFE strategy (Chen, Shao, Wang, B., 2012)

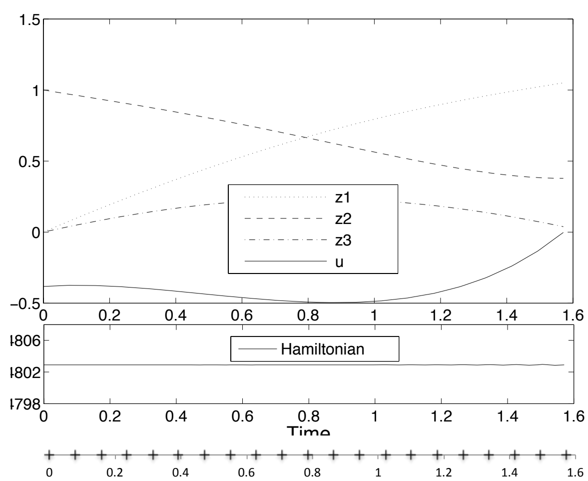
Need to satisfy Coercivity Conditions to guarantee SSOSC and unique sensitivity.



185

Modified Aly-Chan Singular Problem

- 20→ 21 elements
- One OP Solution
- 26 IP Iterations
- 2.5 CPUs
- H = 0.04052572
- Φ = -0.01501906



DAE Optimization Resources

References

Betts, J. T., Practical Methods for Optimal Control Using Nonlinear Programming, SIAM, (2010)

Biegler, L. T., Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Engineering, SIAM, Philadelphia (2010)

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)
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)

187

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 (2007)

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

Biegler, L. T., Nonlinear Programming: Concepts, Algorithms and Applications to Chemical Engineering, SIAM (2010)

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

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.

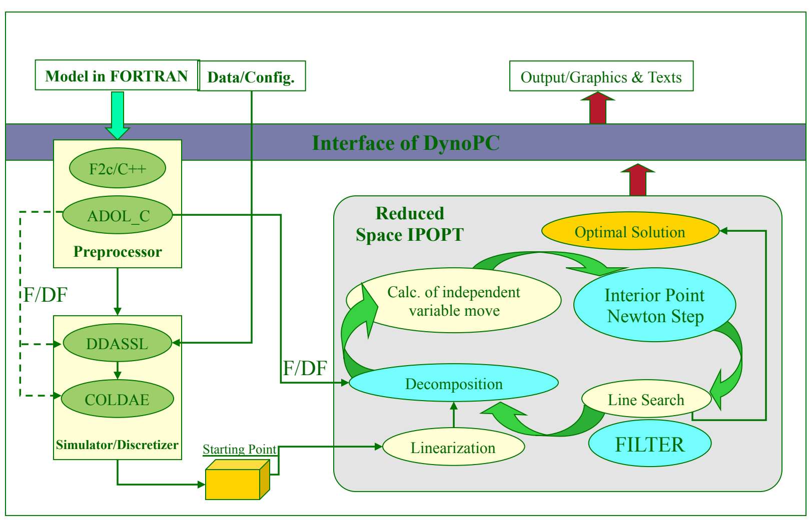
M. Liepelt, K Schittkowski, Optimal control of distributed systems with breakpoints, p. 271 in M. Grötschel, S. Krumke, J. Rambau (eds.), *Online Optimization of Large Systems*, Springer, Berlin (2001)

V. M. Zavala, and L.T.Biegler, "Optimization-Based Strategies for the Operation of Low-Density Polyethylene Tubular Reactors: Nonlinear Model Predictive Control," Computers and Chemical Engineering, 33, pp. 1735-1746 (2009)

See also Biegler homepage

43

DynoPC – Windows Implementation

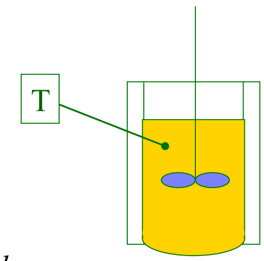


190

Example: Batch Reactor Temperature

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

$$Max \quad b(t_f)$$



$s.t.$

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

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

$$a + b + c = 1$$

Example: Car Problem

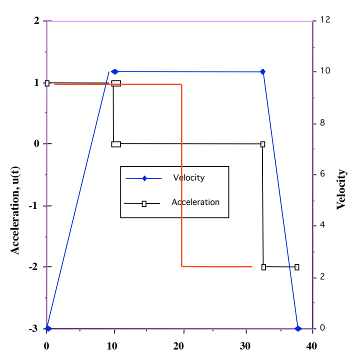
Min $\quad t_f$

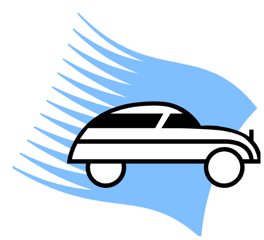
s.t. $\quad z_1' = z_2$

$z_2' = u$

$z_2 \leq z_{max}$

$-2 \leq u \leq 1$





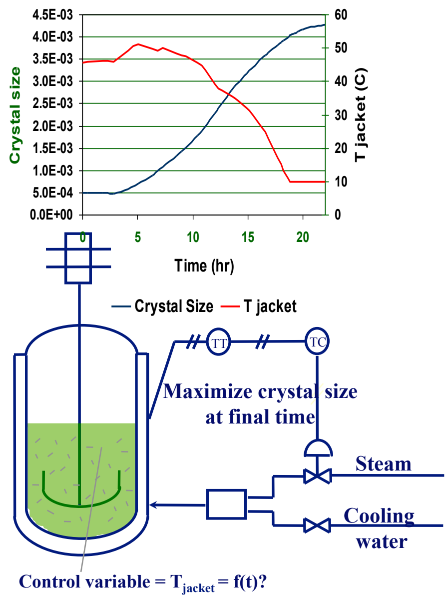
```
subroutine model(nz,ny,nu,np,t,z,dmz,y,u,p,f)
double precision t, z(nz),dmz(nz), y(ny),u(nu),p(np)

double precision f(nz+ny)

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

return
end
```

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
```

193