A Brief GAMS Tutorial for Dynamic Optimization

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

Dynamic Optimization Problem

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

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

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

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

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

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

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

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

t, time
z, differential variables
y, algebraic variables

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

DAE Models in Engineering

Differential Equations

- Conservation Laws (Mass, Energy, Momentum)

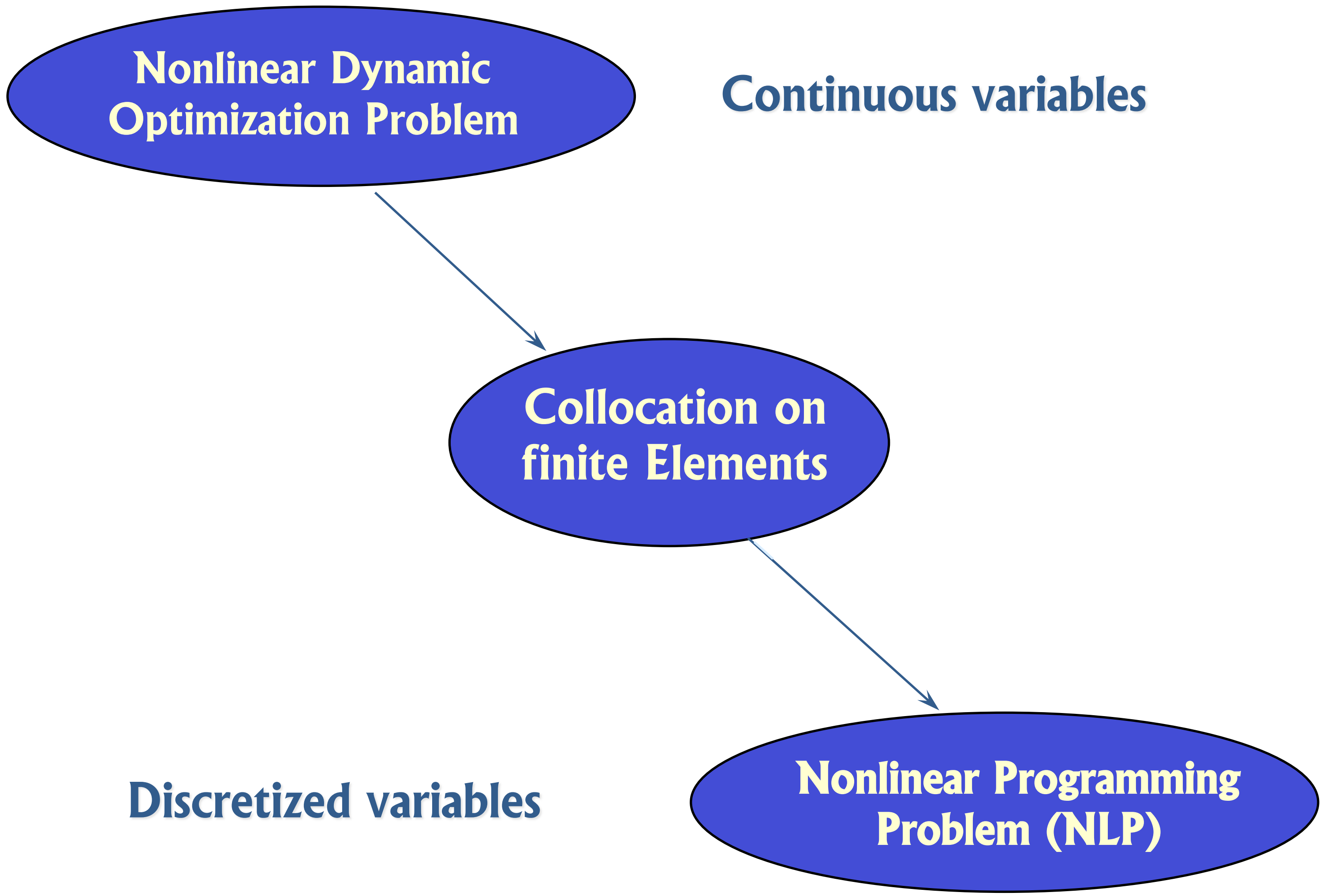
Algebraic Equations

- Constitutive Equations, Equilibrium (physical properties, hydraulics, rate laws)

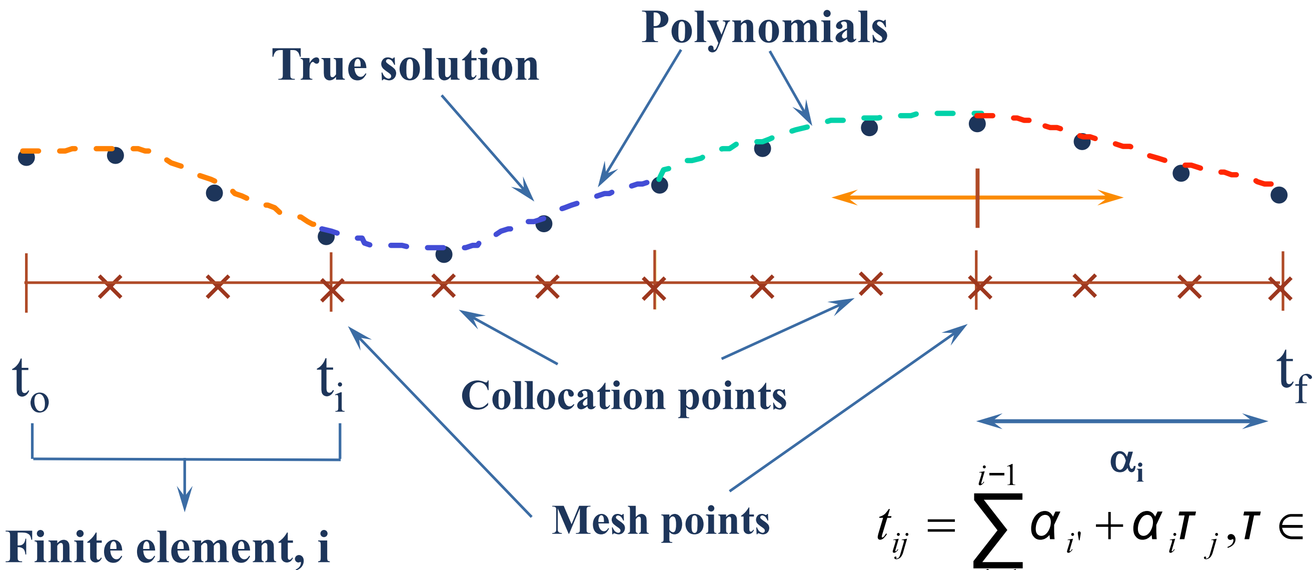
Characteristics

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

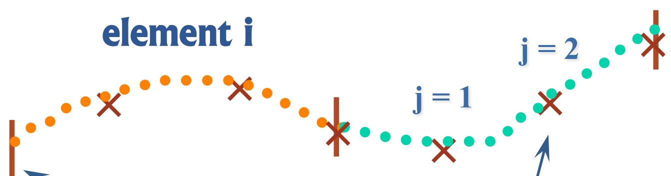
Nonlinear Programming Formulation



Collocation on Finite Elements



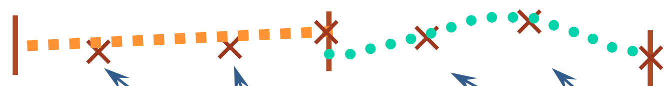
$$t_{ij} = \sum_{i'=1}^{i-1} \alpha_{i'} + \alpha_i \tau_j, \tau \in [0,1]$$



$$z(t) = \sum_{j=0}^{NCOL} \ell_j(\tau) z_{ij}$$

$$\ell_j(\tau) = \prod_{\substack{l=0,\\ l \neq j}}^{NCOL} \frac{\tau - \tau_j}{\tau_l - \tau_j}$$

Differential variables Continuous



$$y(t) = \sum_{j=1}^{NCOL} \bar{\ell}_j(\tau) y_{ij} \qquad u(t) = \sum_{j=1}^{NCOL} \bar{\ell}_j(\tau) u_{ij}$$

$$\bar{\ell}_j(\tau) = \prod_{\substack{l=1,\\ l \neq j}}^{NCOL} \frac{\tau - \tau_j}{\tau_l - \tau_j}$$

Algebraic and Control variables Discontinuous

Nonlinear Programming Problem

$$\min \ \psi\left(z_{i,j}, y_{i,j}, u_{i,j}, p, t_f\right)$$

$$s.t. \quad \sum_{k=0}^{NCOL} \ell_k(\tau_j) z_{ik} = \alpha_i F(z_{ij}, y_{ij}, u_{ij}, p)$$

$$G\left(z_{i,j}, y_{i,j}, u_{i,j}, p\right) = 0$$

$$z_{i+1,0} = \sum_{j=0}^{NCOL} \ell_j(1) z_{ij}$$

$$z_1^{\,o} = z(0), \ z_f = z_{i+1}^{\,0}$$

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

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

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

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

$$i = 1,\ldots NFE; \ \ j = 1,\ldots NCOL$$

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

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

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

Dynamic Optimization: Methods of Solution

- GAMS – requires complex set notation and conditional statements, general formulation with all collocation equations
- AMPL – similar to GAMS, more elegant set notation and conditional statements, general formulation with all collocation equations
- DynoPC – requires only statement of model, data-driven program in Windows, uses IPOPT and ADOL-C

Example: Hicks Reactor Problem

Consider the liquid-phase stirred tank reactor shown in Figure 1. We assume

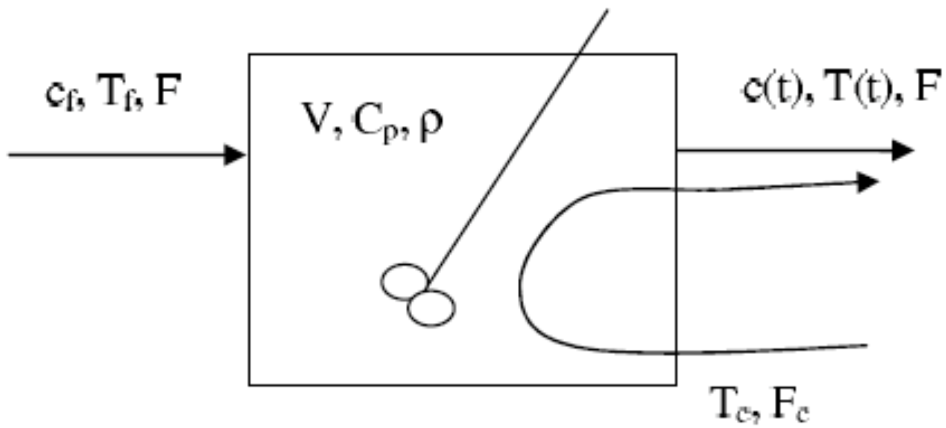


Figure 1: Sketch of Hicks Reactor

constant liquid volume (V), flow (F), density (ρ), heat capacity (C_p), coolant temperature (T_c), coolant flow (F_c) and heat of reaction (ΔH), as well as a first order Arrhenius rate law. Following the notation in the figure, the mass and energy balance for the reactor are given by:

Example: Hicks Reactor Problem

$$V\frac{dc}{dt} = F(c_f - c(t)) - Vk_{10}exp(-E/RT)c(t), \quad c(0) = c_{init} \qquad (1)$$

$$\rho C_p V\frac{dT}{dt} = F\rho C_p(T_f - T(t)) + \Delta H V k_{10}exp(-E/RT)c(t)$$
$$+U(F_c)A_c(T_c - T), \quad T(0) = T_{init} \qquad (2)$$

We define aggregated parameters $\theta = V/F$, $\alpha u = UA_c/(\rho C_p V)$, $y_f = \rho C_p T_f/\Delta H$, $n = E\rho C_p/(R\Delta H)$, $y_c = \rho C_p T_c/\Delta H$, and redefine the states: $c \leftarrow c/c_f$, $T \leftarrow \rho C_p T/\Delta H$ in order to obtain:

$$\frac{dc}{dt} = (1 - c(t))/\theta - k_{10}exp(-n/T)c(t), c(0) = c_{init} \qquad (3)$$

$$\frac{dT}{dt} = (y_f - T(t))/\theta - k_{10}exp(-n/T)c(t) + \alpha u(y_c - T), T(0) = T_{init} \qquad (4)$$

The parameters are given values in the GAMS file below and the optimal control problem is given by:

$$Min \quad \int_0^{t_f} \alpha_1(\bar{c} - c(t))^2 + \alpha_2(\bar{T} - T(t))^2 + \alpha_3(\bar{u} - u(t))^2 dt \qquad (5)$$

$$s.t. \quad (3) - (4) \qquad (6)$$

where $\bar{c}$, $\bar{T}$, $\bar{u}$ are the desired values of the states and input and α_j are the desired weights.

Introduction and define Sets

```
$Title Dynamic Optimization of a Non-Isothermal CSTR

$OFFUPPER
$OFFSYMXREF OFFSYMLIST


* option sysout = off;
* OPTION SOLPRINT = OFF;

* This file implements the dynamic optimization of non-isothermal
* CSTR as proposed by Hicks and Ray and later modified to feature
* multiple steady-state behaviour. Full discretization of the model
* and input, output variables renders the problem as a NLP.
*
* More detail on the model can be found in
*
*Flores Tlacuahuac, A., S. Terrazas Moreno, and L. T. Biegler,
*''On Global Optimization of Highly Nonlinear Dynamic Systems,''
*Industrial and Engineering Chemistry Research, 47, 8, pp 2643 - 2655
*(2008)

Sets    i     number of finite elements             /1*100/
        j     number of internal collocation points /1*3/;
```

```
Alias (j,k);

Scalar  cinit  initial concentration         /0.1367/
        tinit  initial temperature           /0.7293/
        uinit  initial cooling water         /390/
        cdes   initial concentration         /0.0944/
        tdes   final temperature             /0.7766/
        udes   final cooling water flowrate  /340/
        alpha  dimensionless parameter       /1.95e-04/
        alpha1 dimensionless parameter       /1e+06/
        alpha2 dimensionless parameter       /2e+03/
        alpha3 dimensionless parameter       /1e-03/
        k10    rate constant                 /300/
        n                                    /5/
        cf                                   /7.6/
        tf                                   /300/
        tc                                   /290/
        theta                                /20/
        yf                                   /0.3947/
        yc                                   /0.3816/
       time                                 /10/
        point                                /0/
        nfe                                  /100/
        ncp                                  /3/
        slopec
        slopet
        slopeu,
        ii,
        jj,
        point;
```

Carnegie Mellon

Define parameters for collocation

```
Table  a(j,j)  First order derivatives collocation matrix
                 1                  2                  3
     1   0.19681547722366   0.39442431473909   0.37640306270047
     2  -0.06553542585020   0.29207341166523   0.51248582618842
     3   0.02377097434822  -0.04154875212600   0.11111111111111;

Parameter
        cguess(i,j)
        tguess(i,j)
        ttguess(i,j)
        uguess(i,j)
        h(i)        ;
*
* Initial guess of the decision variables
*
point = 0;
slopec = (cdes-cinit)/(nfe*ncp);
slopet = (tdes-tinit)/(nfe*ncp);
slopeu = (udes-uinit)/(nfe*ncp);
for (ii = 1 to nfe,
    for (jj = 1 to ncp,
         point = point+1;
         cguess(i,j) = slopec*point+cinit ;
         tguess(i,j) = slopet*point+tinit ;
         ttguess(i,j) = time*point ;
         uguess(i,j) = slopeu*point+uinit ;
     );
);
h(i) = 1/nfe ;
```

Define Variables and Equations

```
Variables c(i,j)    concentration
          t(i,j)    temperature
          tt(i,j)    time
          u(i,j)    cooling water flowrate
          cdot(i,j)
          tdot(i,j)
          c0(i)
          t0(i)
          tt0(i)
          u0(i)
          phi          objective function ;

 Equations  fobj          criterion definition

          IT, IC, ITT
          FECOLc(i,j)
          FECOLt(i,j)
          FECOLtt(i,j)
          CONc(i)
          CONt(i)
          CONtt(i)
          ODEc(i,j)
          ODEt(i,j);
```

Collocation Model

```
fobj..
phi =e= sum((i,j), h(i)*a(j,'3')*(alpha1*(sqr(cdes-c(i,j)))+alpha2*
        sqr(tdes-t(i,j))+alpha3*sqr(udes-u(i,j))));

FECOLc(i,j)$(ord(i) le nfe).. c(i,j)=e=c0(i)
+time*h(i)*sum(k,a(k,j)*cdot(i,k)) ;

FECOLt(i,j)$(ord(i) le nfe).. t(i,j) =e= t0(i)
+time*h(i)*sum(k,a(k,j)*tdot(i,k)) ;

FECOLtt(i,j)$(ord(i) le nfe).. tt(i,j) =e= tt0(i)
+time*h(i)*sum(k,a(k,j)) ;

CONc(i)$(ord(i) gt 1 and ord(i) le nfe)..
c0(i) =e= c0(i-1) + time*h(i-1)*sum(j, cdot(i-1,j)*a(j,'3'));

CONt(i)$(ord(i) gt 1 and ord(i) le nfe)..
t0(i) =e= t0(i-1) + time*h(i-1)*sum(j, tdot(i-1,j)*a(j,'3'));

CONtt(i)$(ord(i) gt 1 and ord(i) le nfe)..
tt0(i) =e= tt0(i-1) + time*h(i-1)*sum(j, a(j,'3'));

ODEc(i,j)$(ord(i) le nfe).. cdot(i,j) =e= (1-c(i,j))/theta
-k10*exp(-n/t(i,j))*c(i,j) ;

ODEt(i,j)$(ord(i) le nfe).. tdot(i,j) =e= (yf-t(i,j))/theta
+k10*exp(-n/t(i,j))*c(i,j)-alpha*u(i,j)*(t(i,j)-yc) ;
IC.. c0('1') =e= cinit;
       IT.. t0('1') =e= tinit;
       ITT.. tt0('1') =e= 0;
```

Initialization and Solution

```
Model hicks /all/;
c.lo(i,j) = 0; c.up(i,j) = 1;
t.lo(i,j) = 0.1; t.up(i,j) = 1;
u.lo(i,j) = 0; u.up(i,j) = 500;
c0.lo(i) = 0; c0.up(i) = cinit ;
t0.lo(i) = 0.1; t0.up(i)= 1;
c.l(i,j)  = cguess(i,j);
t.l(i,j)  = tguess(i,j);
tt.l(i,j)  = ttguess(i,j);
u.l(i,j)  = uguess(i,j);
c0.l(i)   = cinit ;
t0.l(i)   = tinit ;
u0.l(i)   = uinit ;
cdot.l(i,j) = 1 ;
tdot.l(i,j) = 1 ;

option nlp = coinipopt;
* option nlp = baron;
* option nlp = conopt;
* option nlp = minos;
*hicks.optfile = 1;

Solve hicks minimizing phi using nlp;

display tt0.l, c0.l, t0.l;
* option nlp = conopt;
* Solve hicks minimizing phi using nlp;
```