



A Brief GAMS Tutorial

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



To access GAMS

The computers in the lounge have all been provisioned with GAMS 22.7. GAMS is accessible from the command prompt.

The form of the accounts are as follows:

Username: capd## (01-20)

Password: scpwd

Open folder “L. Biegler – CAPD 2012” on Desktop

Open CAPD

Chemical ENGINEERING

GAMS Optimization Background

Problem Statement

$$\begin{aligned} & \text{Min } F(x, y) \\ & \text{s.t. } h(x, y) = 0 \\ & \quad g(x, y) \leq 0 \\ & \quad x \in \mathbb{R}^{nx}, y \in \{0, 1\}^{ny} \end{aligned}$$

Problem Classes

LP:	CPLEX, XPRESS, ZOOM
MILP:	CPLEX, XPRESS, ZOOM
NLP:	CONOPT, IPOPT, KNITRO, MINOS
MINLP:	DICOPT, BONMIN

Approach

- Leverage powerful existing solvers
- Less emphasis on building algorithms
- More emphasis on model formulation and refinement

3

Chemical ENGINEERING

GAMS Structure

```

    graph LR
      A[GAMS Input File (name.gms)] --> B((GAMS Compilation And Expansion))
      B --> C[GAMS Output File (name.lst)]
      D[Optimization Solver] <--> B
  
```

The diagram illustrates the GAMS workflow. It starts with a 'GAMS Input File (name.gms)' which feeds into a central 'GAMS Compilation And Expansion' process. This process then produces a 'GAMS Output File (name.lst)'. Additionally, the 'GAMS Compilation And Expansion' process interacts bidirectionally with an 'Optimization Solver'.

- Common language across platforms, versions
- Easy to apply (and compare) different solvers
- Syntax and logic checking prior to solver execution
- Common output file for interpretation
- Automatic differentiation of model
- Closed system, external procedures hard to implement
- Not much user interaction with solvers

4



Input File Syntax (name.gms)

Essential Parts of File

\$TITLE *State problem title here*

VARIABLES *list variables here*

EQUATIONS *list equations here*

.....

Problem Statement – state objective and constraint functions

.....

MODEL *identify all equations in model_name*

SOLVE *model_name* USING *problem_type* MINIMIZING *objective_variable*

Some Syntax Rules

All statements end with semicolons;

=G=, =L=, =E=, conventions

X.LO, X.UP, X.L, X.M qualifiers for variables

Use these to bound variables, except for *objective_variable*

Comments denoted with ‘*’

5



Introductory Example

$$\text{Min } x_1^2 + x_2^2 + x_3^2$$

$$\text{s.t. } x_2 - x_3 \geq 0$$

$$x_1 - x_3 \geq 0$$

$$x_1 - x_2^2 + x_1^2 - 4 = 0$$

$$0 \leq x_1 \leq 5$$

$$0 \leq x_2 \leq 3$$

$$0 \leq x_3 \leq 3$$

$x^0 = [4, 2, 2]$

Input in test.gms →

See output in test.lst

```

$title Test Problem
$offsymxref
$offsymlist
* Problem 8.26 in Reklaitis et al (1983)
Variables z;
Positive Variables x1, x2, x3;
Equations con1, con2, con3, obj;

```

```

con1.. x2-x3 =G= 0;
con2.. x1-x3 =G= 0;
con3.. x1-x2**2 + x1**2 - 4 =E= 0;
obj.. z =e= sqrt(x1)+sqrt(x2)+sqrt(x3);

```

```

* upper bounds
x1.up = 5;
x2.up = 3;
x3.up = 3;

```

```

* initial point
x1.l = 4;
x2.l = 2;
x3.l = 2;

```

```

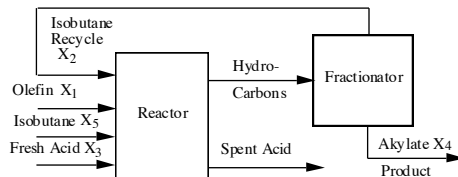
Model test /all/;
Solve test using NLP minimizing z;

```

6

Alkylation Problem

Consider the alkylation process shown below from Bracken & McCormick (1968)



X1 = Olefin feed (barrels per day)
X2 = Isobutane recycle (barrels per day)

X6 = Acid strength (weight percent)
X7 = Motor octane number of alkylate

The alkylation is derived from simple mass balance relationships and regression equations determined from operating data.

The objective function to be maximized is the profit (\$/day)

$$OBJ = 0.063 * X4 * X7 - 5.04 * X1 - 0.035 * X2 - 10 * X3 - 3.36 * X5$$

7

Alkylation Problem – Input File

```

$ TITLE ALKYLATION PROBLEM FROM GINO USER'S MANUAL
$ OFFSYMXREF
$ OFFSYMLIST
OPTION LIMROW=0;
OPTION LIMCOL=0;

POSITIVE VARIABLES X1,X2,X3,X4,X5,X6,X7,X8,X9,X10;
VARIABLE OBJ;
EQUATIONS E1,E2,E3,E4,E5,E6,E7,E8;

E1..X4=E=X1*(1.12+13167*X8-0.0067*X8**2);
E2..X7=E=86.35+1.098*X8-0.038*X8**2+0.325*(X6-89.);
E3..X9=E=35.82-0.222*X10;
E4..X10=E=3*X7-133;
E5..X8*X1=E=X2+X5;
E6..X5=E=1.22*X4-X1;
E7..X6*(X4*X9+1000*X3)=E=98000*X3;

E8.. OBJ =E= 0.063*X4*X7-5.04*X1-0.035*X2-10*X3-3.36*X5;

X1.UP = 2000.;
X2.UP = 16000.;
X3.UP = 120.;

X4.UP = 5000.;
X5.UP = 2000.;
X6.LO = 85.;
X6.UP = 93.;
X7.LO = 90;
X7.UP = 95;
X8.LO = 3.;
X8.UP = 12.;
X9.LO = 1.2;
X9.UP = 4.;
X10.LO = 145.;
X10.UP = 162.;
X1.L = 1745;
X2.L = 12000;
X3.L = 110;
X4.L = 3048;
X5.L = 1974;
X6.L = 89.2;
X7.L = 92.8;
X8.L = 8;
X9.L = 3.6;
X10.L = 145;

MODEL ALKY/ALL;
SOLVE ALKY USING NLP MAXIMIZING OBJ;
  
```

8

Chemical ENGINEERING

Refinery Scheduling: Additional Features of GAMS: Sets, Index Notation, External Calcs

```

    graph LR
      C1 --> FC[Fuel Chain]
      C2 --> FC
      C3 --> FC
      C4 --> LC[Lube Chain]
      FC --> Gasoline
      FC --> HO[Heating Oil]
      FC --> JF[Jet Fuel]
      LC --> LO[Lube Oil]
  
```

Product Yields for Crudes

	1	2	3	4-fuel	4-lube
gasoline	0.6	0.5	0.3	0.4	0.4
heat-oil	0.2	0.2	0.3	0.3	0.1
jet-fuel	0.1	0.2	0.3	0.2	0.2
lube-oil	0.0	0.0	0.0	0.0	0.2;

9

Chemical ENGINEERING

Refinery Scheduling: Problem Data

```

$title Refinery Scheduling
$offupper
$offsymxref $offsymlist

*OPTION SOLPRINT = OFF;

* Define index sets
SETS C Crudes /1*3, 4-fuel, 4-lube/
     P Products /gasoline, heat-oil, jet-fuel, lube-oil/

*Define and initialize the problem data
PARAMETER CSUPPLY(C) Crude oil supply kbbbl per wk / 1 100.0, 2 100.0, 3 100.0, 4-fuel 200.0,
4-lube 200.0/
CCOST(C) Crude oil costs in $ per bbl /1 15.0, 2 15.0, 3 15.0, 4-fuel 25.0, 4-lube 25.0/
PDEMAND(P) Maximum product demands, kbbbl per wk /gasoline 170.0, heat-oil 85.0, jet-fuel
85.0, lube-oil 20.0/
PVALUE(P) Product Values in $ per bbl / gasoline 45.0, heat-oil 30.0, jet-fuel 15.0, lube-oil 60.0/
OCOST(C) Crude operating costs in $ per bbl / 1 5.0, 2 8.5, 3 7.5, 4-fuel 3.0, 4-lube 2.50/
TCOST(C) Total costs: crude plus operating;
TCOST(C) = CCOST(C) + OCOST(C);

TABLE YIELDS(P, C) Yields of products for crudes
      1  2  3  4-fuel 4-lube
gasoline 0.6 0.5 0.3 0.4 0.4
heat-oil 0.2 0.2 0.3 0.3 0.1
jet-fuel 0.1 0.2 0.3 0.2 0.2
lube-oil 0.0 0.0 0.0 0.0 0.2;
  
```

10



Refinery Scheduling: Problem Statement

* Define the optimization variables

VARIABLES X(C) Crude oils used in kbbl per week
Q(P) Amounts of products produced in kbbl
X4 Total amount of crude 4 used in kbbl
PROFIT Total profit from product sales in k\$;

POSITIVE VARIABLES X, X4, Q;

* Define constraints and objective function

EQUATIONS OBJFUN Objective function to be maximized

CRUDE4 Total crude 4 usage
PRODUCTION(P) Amounts of products produced;
OBJFUN.. PROFIT =E= SUM(P, Q(P)*PVALUE(P)) - SUM(C, TCOST(C)*X(C));
PRODUCTION(P).. Q(P) =E= SUM(C, YIELDS(P,C)*X(C));
CRUDE4.. X4 =E= X("4-fuel") + X("4-lube");

* Define upper and lower bounds

* Upper bounds on amounts of product produced from their maximum demands

Q.UP(P) = PDEMAND(P);

* Upper bounds on crude oil usages from their supplies

X.UP(C) = CSUPPLY(C);

X4.UP = CSUPPLY("4-fuel");

* Define model and solve

MODEL SCHEDULE /ALL/;

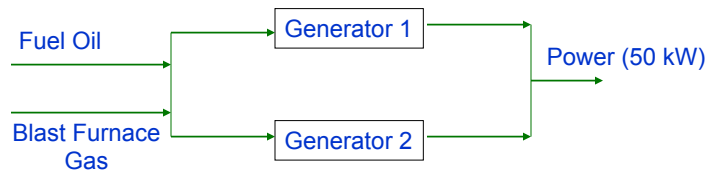
SOLVE SCHEDULE USING LP MAXIMIZING PROFIT;

DISPLAY X.L, Q.L, PROFIT.L;

11



Power Generation via Fuel Oil



Minimize Fuel Oil Consumption

$$\text{Fuel (oil or BFG)} = A_0 + A_1 * \text{Power} + A_2 * (\text{Power})^2$$

Coefficients in the fuel consumption equations

	A_0	A_1	A_2
gen ₁ (oil)	1.4609	.15186	.00145
gen ₁ (gas)	1.5742	.16310	.001358
gen ₂ (oil)	0.8008	.20310	.000916
gen ₂ (gas)	0.7266	.22560	.000778;

12



Power Generation via Fuel Oil: Problem Data

```

$title Power Generation via Fuel Oil
$offupper
$offsymxref $offsymlist

*option solprint = off;

* Define index sets
sets G Power Generators /gen1*gen2/
    F Fuels /oil, gas/
    K Constants in Fuel Consumption Equations /0*2/;

* Define and Input the Problem Data
table A(G,F,K) Coefficients in the fuel consumption equations
      0      1      2
gen1.oil  1.4609  .15186  .00145
gen1.gas  1.5742  .16310  .001358
gen2.oil  0.8008  .20310  .000916
gen2.gas  0.7266  .22560  .000778;

parameter Pmax(G) Maximum power outputs of generators /
    GEN1 30.0, GEN2 25.0/;
parameter Pmin(G) Minimum power outputs of generators /
    GEN1 18.0, GEN2 14.0/;
scalar GASSUP Maximum supply of BFG in units per h /10.0/
    PREQ Total power output required in MW /50.0/;

```

13



Power Generation via Fuel Oil: Problem Statement

```

* Define optimization variables
variables P(G) Total power output of generators in MW
          X(G, F) Power outputs of generators from specific fuels
          Z(F) Total Amounts of fuel purchased
          OILPUR Total amount of fuel oil purchased;
positive variables P, X, Z;

* Define Objective Function and Constraints
equations TPOWER Required power must be generated
          PWR(G) Power generated by individual generators
          OILUSE Amount of oil purchased to be minimized
          FUELUSE(F) Fuel usage must not exceed purchase;
TPOWER.. SUM(G, P(G)) =G= PREQ;
PWR(G).. P(G) =E= SUM(F, X(G,F));
FUELUSE(F).. Z(F) =G= SUM((K,G), a(G,F,K)*X(G,F)**(ORD(K)-1));
OILUSE.. OILPUR =E= Z("OIL");

* Impose Bounds and Initialize Optimization Variables
* Upper and lower bounds on P from the operating ranges
P.UP(G) = Pmax(G);
P.LO(G) = Pmin(G);
* Upper bound on BFG consumption from GASSUP
Z.UP("gas") = GASSUP;
* Specify initial values for power outputs
P.L(G) = .5*(Pmax(G)+Pmin(G));

* Define model and solve
model FUELOIL /all/;
solve FUELOIL using NLP minimizing OILPUR;

display X.L, P.L, Z.L, OILPUR.L;

```

14



Suggested GAMS Exercises

- Refinery Scheduling – make product demands lower bound requirements
 - Maximize profit
 - Minimize operating cost
- Fuel Oil – Restrict fuel oil supply to 10 ton/h, purchase BFG.
 - Minimize BFG purchased
 - Minimize BFG and Fuel Oil purchased
- Alkylation – Add relaxation variables between bounds and reformulate problem

15



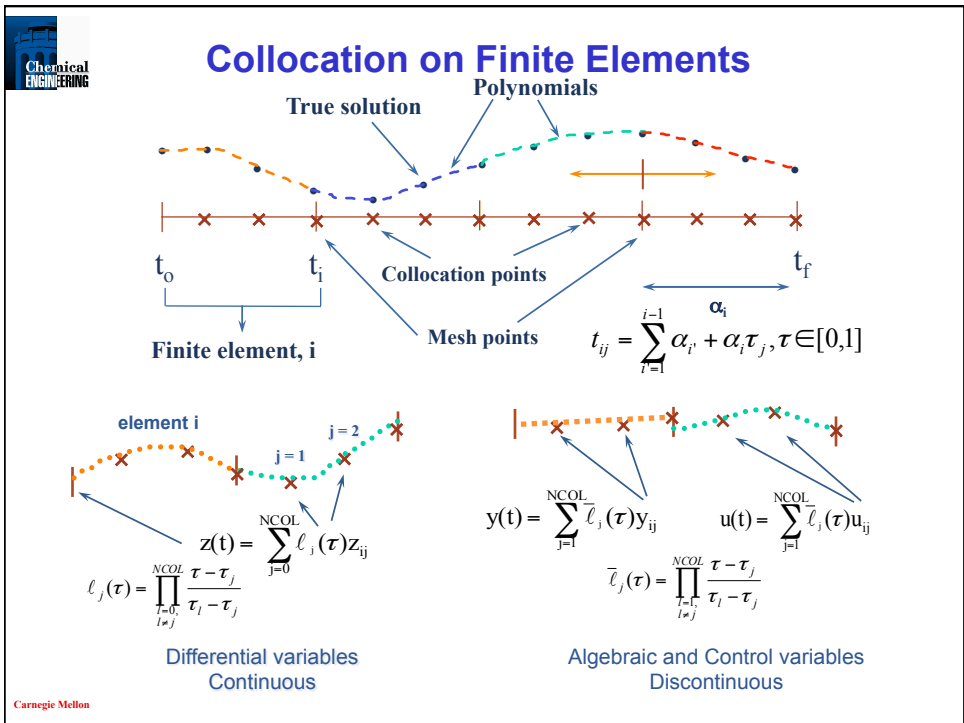
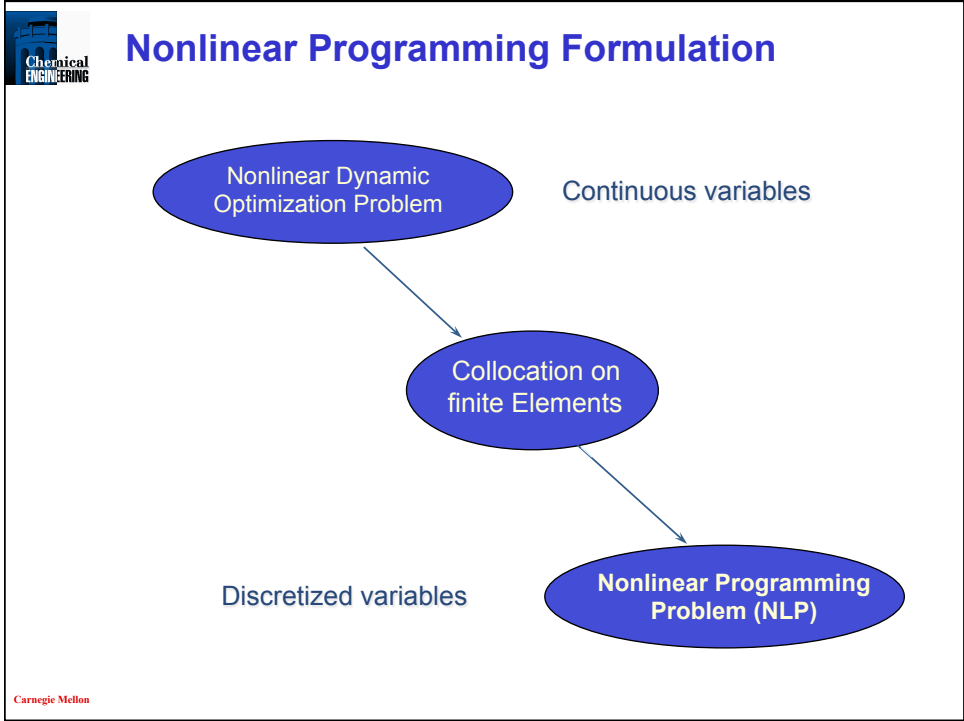
Dynamic Optimization Problem

$$\begin{aligned} \min \quad & \psi(z(t), y(t), u(t), p, t_f) \\ \text{s.t.} \quad & \frac{dz(t)}{dt} = F(z(t), y(t), u(t), t, p) \\ & G(z(t), y(t), u(t), t, p) = 0 \\ & z^o = z(0) \\ & z' \leq z(t) \leq z'' \\ & y' \leq y(t) \leq y'' \\ & u' \leq u(t) \leq u'' \\ & p' \leq p \leq p'' \end{aligned}$$

t , time
 z , differential variables
 y , algebraic variables

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

Carnegie Mellon

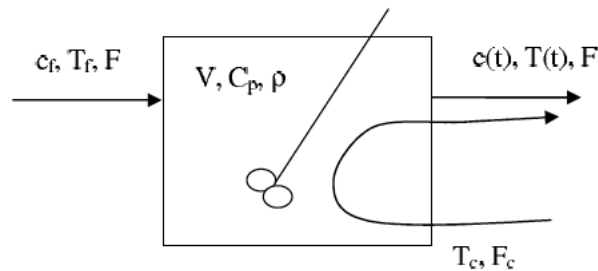


Nonlinear Programming Problem

$$\begin{aligned} & \min \psi(z_{i,j}, y_{i,j}, u_{i,j}, p, t_f) \\ \text{s.t. } & \sum_{k=0}^{NCOL} \ell_k(\tau_j) z_{ik} = \alpha_i F(z_{ij}, y_{ij}, u_{ij}, p) \\ & G(z_{i,j}, y_{i,j}, u_{i,j}, p) = 0 \\ & z_{i+1,0} = \sum_{j=0}^{NCOL} \ell_j(1) z_{ij} \\ & z_1^o = z(0), z_f = z_{i+1}^o \\ & z_i^l \leq z_{i,j} \leq z_i^u \\ & y_{i,j}^l \leq y_{i,j} \leq y_{i,j}^u \\ & u_{i,j}^l \leq u_{i,j} \leq u_{i,j}^u \\ & p^l \leq p \leq p^u \\ & i = 1, \dots, NFE; j = 1, \dots, NCOL \end{aligned}$$

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} f(x) \\ \text{s.t. } & c(x) = 0 \\ & x^L \leq x \leq x^u \end{aligned}$$

Optimal Control for Dynamic CSTR (Hicks and Ray, 1971)



$$\begin{aligned} \text{Min } & \int_0^{t_f} \alpha_1 (\bar{c} - c(t))^2 + \alpha_2 (\bar{T} - T(t))^2 + \alpha_3 (\bar{u} - u(t))^2 dt \\ \frac{dc}{dt} &= (1 - c(t))/\theta - k_{10} \exp(-n/T) c(t), c(0) = c_{init} \\ \frac{dT}{dt} &= (T_f - T(t))/\theta - k_{10} \exp(-n/T) c(t) + \alpha u (T_c - T), T(0) = T_{init} \end{aligned}$$



Runge-Kutta Collocation Representation

$$z^K(t) = z_{i-1} + h_i \sum_{j=1}^K \Omega_j(\tau) \dot{z}_{ij}$$

$$\Omega_j(\tau) = \int_0^\tau \bar{\ell}_j(\tau') d\tau', \quad t \in [t_{i-1}, t_i], \quad \tau \in [0, 1].$$

$$\frac{dz^K(t)}{d\tau} = \sum_{j=1}^K \bar{\ell}_j(\tau) \dot{z}_{ij}, \quad \bar{\ell}_j(\tau) = \prod_{k=1, k \neq j}^K \frac{(\tau - \tau_k)}{(\tau_j - \tau_k)}.$$

$$\dot{z}_{ik} = f(z_{ik}, t_{ik})$$

$$z_{ik} = z_{i-1} + h_i \sum_{j=1}^K \Omega_j(\tau_k) \dot{z}_{ij}, \quad k = 1, \dots, K$$

$$z_i = z_{i-1} + h_i \sum_{j=1}^K \Omega_j(1) \dot{z}_{ij}, \quad i = 1, \dots, N-1$$

Carnegie Mellon



CSTR Example with Orthogonal Collocation

$$\min \sum_{i=1}^{NFE} \sum_{j=1}^{NCP} W_j \left[\alpha_1 (C_{ij} - C_{des})^2 + \alpha_2 (T_{ij} - T_{des})^2 + \alpha_3 (U_{ij} - U_{des})^2 \right]$$

st

$$C_{ij} = C_i^o + h_i \theta_d \sum_{k=1}^{NCP} A_{kj} \dot{C}_{ik} \quad i = 1, \dots, N, \quad j = 1, \dots, K$$

$$T_{ij} = T_i^o + h_i \theta_d \sum_{k=1}^{NCP} A_{kj} \dot{T}_{ik} \quad i = 1, \dots, N, \quad j = 1, \dots, K$$

$$\dot{C}_{ij} = \frac{1 - C_{ij}}{\theta_d} - k_{10} \exp\left(-\frac{N}{T_{ij}}\right) C_{ij} \quad i = 1, \dots, N, \quad j = 1, \dots, K$$

$$\dot{T}_{ij} = \frac{y_j - T_{ij}}{\theta_d} + k_{10} \exp\left(-\frac{N}{T_{ij}}\right) C_{ij} - \alpha U_{ij} (T_{ij} - y_c) \quad i = 1, \dots, N, \quad j = 1, \dots, K$$

$$C_i^o = C_{i-1}^o + h_{i-1} \theta_d \sum_{k=1}^{NCP} A_{k,NCP} \dot{C}_{i-1,k} \quad i = 2, \dots, N$$

$$T_i^o = T_{i-1}^o + h_{i-1} \theta_d \sum_{k=1}^{NCP} A_{k,NCP} \dot{T}_{i-1,k} \quad i = 2, \dots, NFE$$

$$C_{NFE,NCP} = C_{des}$$

$$T_{NFE,NCP} = T_{des}$$

$$U_{NFE,NCP} = U_{des}$$

$$U_L \leq U_{ij} \leq U_H \quad i = 1, \dots, NFE, \quad j = 1, \dots, NCP$$

Carnegie Mellon



Creating the GAMS File

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	0.1111111111111111;

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))*c(i,k);

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

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

$$z_{ik} = z_{i-1} + h_i \sum_{j=1}^K \Omega_j(\tau_k) \dot{z}_{ij}, \quad k = 1, \dots, K$$

Carnegie Mellon



Creating the GAMS File

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	0.1111111111111111;

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))*c(i,k);

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

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

$$z_{ik} = z_{i-1} + h_i \sum_{j=1}^K \Omega_j(\tau_k) \dot{z}_{ij}, \quad k = 1, \dots, K$$

Carnegie Mellon



Creating the GAMS File

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	0.1111111111111111;

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'));

$$z_i = z_{i-1} + h_i \sum_{j=1}^K \Omega_j(1) \dot{z}_{ij}, \quad i = 1, \dots, N - 1$$

Carnegie Mellon



Creating the GAMS File

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	0.1111111111111111;

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);

$$\dot{z}_{ik} = f(z_{ik}, t_{ik})$$

Carnegie Mellon



Creating the GAMS File

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/, nfe /100/, ncp /3/
.....

Carnegie Mellon

h(i) = 1/nfe ;

Variables

c(i,j) concentration
t(i,j) temperature
tt(i,j) time
u(i,j) cooling water flowrate
c_{dot}(i,j), t_{dot}(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);

**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)))));**