Computer Laboratories: *Mathematical Formulation and Implementation in GAMS*

1

S. Vitali
Charles University

GAMS

- General Algebraic Modeling System:
  - language
  - set of solvers
- sources:
  - R.E. Rosenthal "A GAMS Tutorial" in A. Brooke, D.Kendrick, A. Meeraus, R. Raman, *Gams a User's Guide*, GAMS Development Corporation, 1998
  - www.gams.com
  - many books

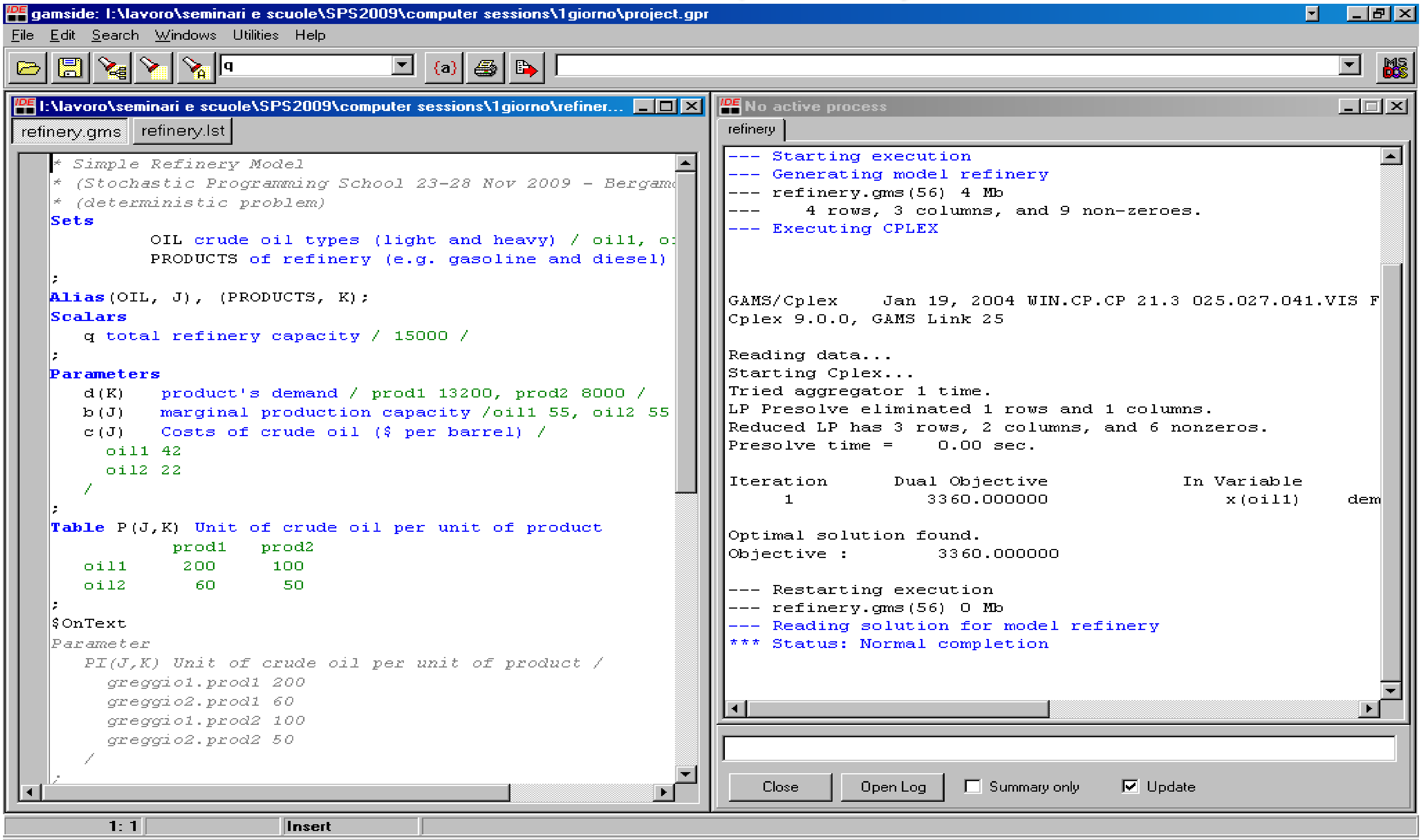
What You Need

- editor or integrated development environment (IDE)
- GAMS modules:
  - GAMS base (compiler)
  - solver (specific for your kind of problems: LP, NLP, …)
- GAMS licence:
  - demo (student) or full for your solver

Where to Get

- You can download the full version (except the licence) directly from the website:
  - www.gams.com
- or buy a CD to your office

Integrated Development Environment (IDE)



Language Essentials

- GAMS language is NOT case-sensitive
- All the statements end with a semicolon (;)
- GAMS statements may be laid out typographically in almost any style that is appealing to the user
- Multiple lines per statement, embedded blank lines, and multiple statements per line are allowed
- An entity cannot be referenced before it is declared to exist

Basic Components

- Comments
- Sets
- Input Data
- Intermediate Computations
- Variables
- Equations
- Models
- Solving
- Output

Agenda

- We overview the GAMS language through three simple optimization problems:

1. Deterministic program of a simple refinery model
2. Simple asset-liability (ALM) model
3. Four-stages ALM stochastic programming model

Agenda

- We overview the GAMS language through three simple optimization problems:

1. Deterministic program of a simple refinery model
2. Simple asset-liability (ALM) model
3. Four-stages ALM stochastic programming model

Simple Refinery Model

$$\min_{\mathbf{x}} \Gamma = \mathbf{c}^\top \mathbf{x} \qquad (1)$$

subject to

$$\mathbf{b}^\top \mathbf{x} \leq q \qquad (2)$$

$$\Pi^\top \mathbf{x} \geq \mathbf{d} \qquad (3)$$

Γ Total costs of crude oil consumption
$\mathbf{x}$ Unit of crude oil consumed in the production
$\mathbf{c}$ Costs of crude oil ($ per barrel)
$\mathbf{b}$ Marginal production capacity
q Total refinery capacity
Π Unit of crude oil per unit of product
$\mathbf{d}$ Total demand of each product

Simple Refinery Model

Input Data

- Type of crude oil (light and heavy), $j = 1, 2$.
- Products of refinery (e.g. gasoline and diesel), $k = 1, 2$.
- Total refinery capacity, $q = 15000$.
- Product's demand, $\mathbf{d}^\top = [13200 \;\; 8000]$.
- Marginal production capacity, $\mathbf{b}^\top = [55 \;\; 55]$.
- Costs of crude oil (\$ per barrel), $\mathbf{c}^\top = [42 \;\; 22]$.
- Unit of crude oil per unit of product, $\Pi = \begin{bmatrix} 200 & 100 \\ 60 & 50 \end{bmatrix}$.

Comments

- Specific lines for comments with '*' at the beginning of a new line:

```
* this is a comment
```

- and in between a statement, when available, e.g.:

```
Set I this is the comment of may
  first set / 1, 2, 3 /;
```

Sets

- Set (or Sets) statement defines a set and its elements:

```
Set
OIL crude oil types (light and heavy)
  / oil1, oil2 /
PRODUCTS of refinery (e.g. gasoline
  and diesel) / prod1*prod2 /
;
```

don't forget it

Input Data

- **Scalar** defines one-value parameter:

```
Scalars q total refinery capacity / 15000 /;
```

- **Parameter** defines any kind of array:

```
Parameters d(K) product's demand /
  prod1 13200, prod2 8000 /;
```

- **Table** reduces the domains of two-dimensional arrays:

```
Table P(J,K) Unit of crude oil per unit
  of product
              prod1     prod2
   oil1        200       100
   oil2         60        50
;
```

at least one char. must match!

Variables

- Decisional variables, intermediate variables and output variables are declared with `Variable` statement, eventually proceeded by **`Positive`**, **`Negative`**, **`Binary`** or **`Integer`**:

```
Positive Variables
   x(J)   Unit of crude oil consumed in
  the production

;
Variables
   gamma Total costs of crude oil
  consumption

;
```

Equations

- The equation require a declaration and an implementation. The declaration is:

```
Equations
   objective objective function
   capacity  refinery capacity
  constraint
   demand    demand of product
  constraints
;
```

- Declaring equations you can specify the set in which it is defined. GAMS will check it in the implementation

Equations

- The implementation of an equation uses a specific syntax:

```
objective..
  gamma =E= Sum(J, c(j) * x(j) );

capacity ..
  Sum(J, b(j) * x(j) ) =L= q;

demand(K)..
  Sum(J, P(j,k)*x(j) ) =G= d(k);
```

separates eq. name from its implementation

$\forall k \in K$

Models

- The statement `Model` lists all the equations used in the model
- The list can be substituted by the keyword `All` whether you want to consider all the equations previously defined

```
Model refinery oil refinery model
  / ALL /;
```

- This statement doesn't distinguish between objective function and constraints

Solving the Problem

- The statement **`Solve`** specifies
  - when to solve a defined model
  - which direction (max or min) to take
  - which is the "output" variable, i.e. the objective function
  - which class of solver you need to solve the model: linear programming (LP), non-linear programming (NLP), mixed-integer (MIP), non-linear with discontinuous derivatives (DNLP), relaxed MIP (RMIP), mixed integer non-linear (MINLP), etc.

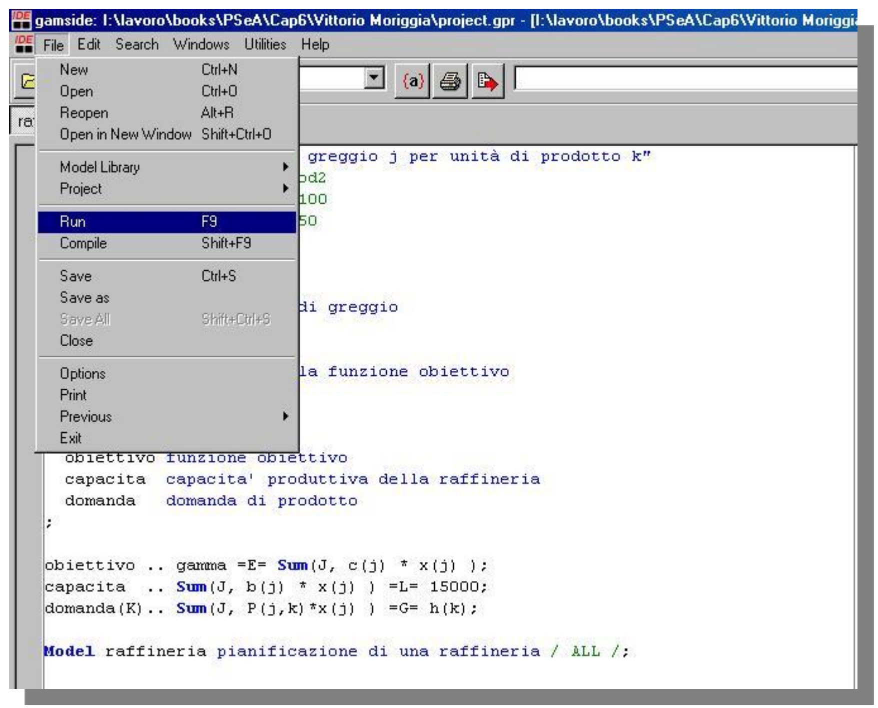
```
Solve refinery Minimizing gamma Using
  LP;
```

Solving the Problem

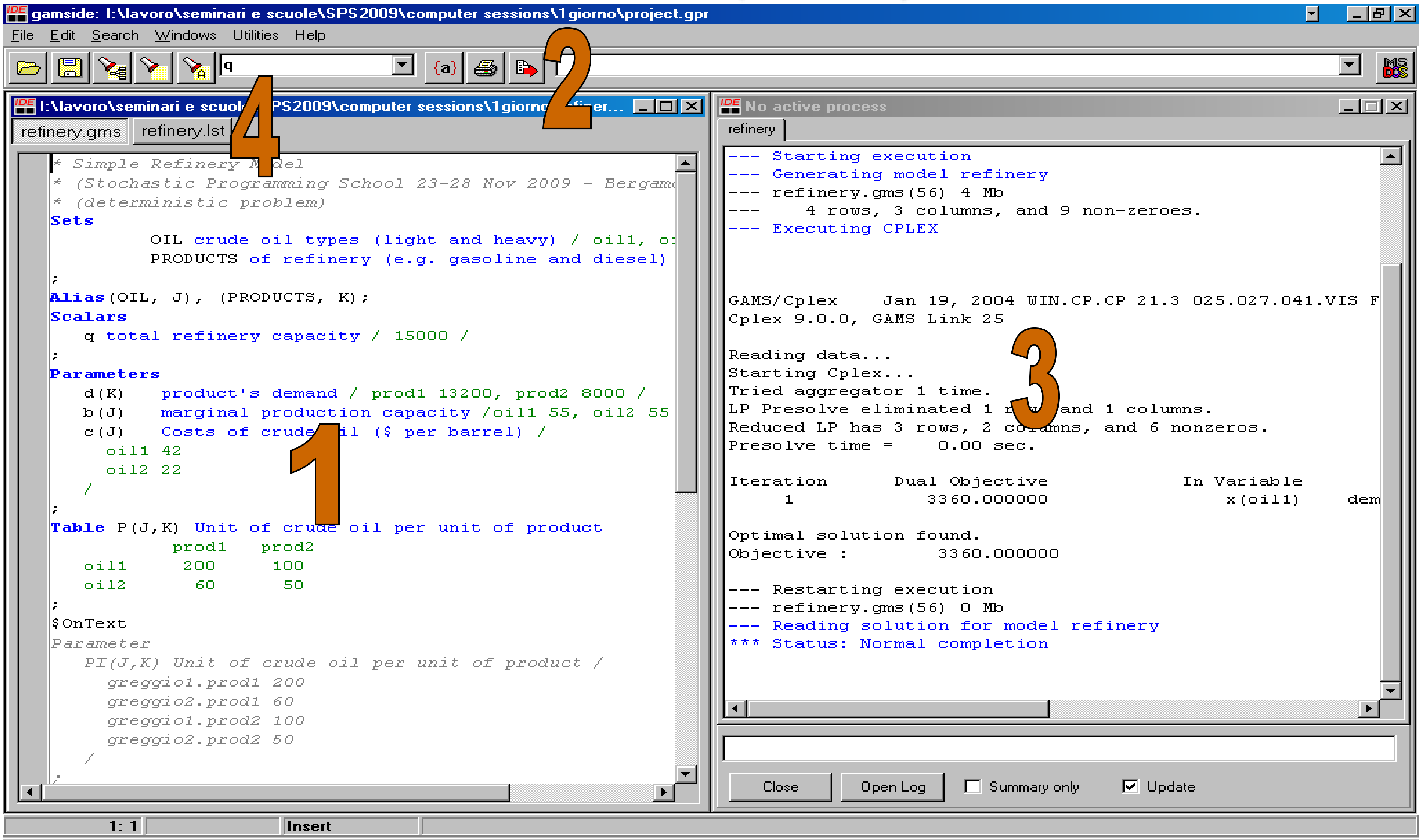
- Once the .gms file is complete you must compile it using both:
  - the command line: `C>gams refinery.gms`

or

  - the IDE (F9 Run):



Integrated Development Environment (IDE)



Agenda

- We overview the GAMS language through three simple optimization problems:

1. Deterministic program of a simple refinery model
2. Simple asset-liability (ALM) model
3. Four-stages ALM stochastic programming model

Simple ALM Model

$$\max_{\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{W}} \mathbb{E}U = \sum_s p_s \left(\mathbf{q}_{T,s}^\top \cdot \mathbf{x}_{T,s} + \mathbf{W}_{T,s}\right) \quad (4)$$

subject to

$$\mathbf{x}_{t+1,s} = \mathbf{x}_{t,s} + \mathbf{z}_{t,s} - \mathbf{y}_{t,s} \qquad \forall s, t < T \quad (5)$$

$$\mathbf{W}_{t+1,s} = \mathbf{W}_{t,s} + \mathbf{q}_t^{s\top} \mathbf{y}_{t,s} - \left(\mathbf{q}_t^s + b\right)^\top \cdot \mathbf{z}_{t,s} \qquad \forall s, t > 0 \quad (6)$$

p_s probability of scenario $s \in S$, such that $\sum_s p_s = 1$;
$\mathbf{q}_{t,s}$ *tel-quel* (fair) sell price;
$\mathbf{x}_{t,s}$ portfolio composition at time t under scenario s;
$\mathbf{W}_t^s$ amount of wealth in cash at time t under scenario s;
$\mathbf{y}_{t,s}$ selling strategy at time t under scenario s;
$\mathbf{z}_t^s$ buying strategy at time t under scenario s;
b bid-ask spread.

Simple ALM Model

Input Data

- Time horizon, $t = 0, \ldots, 3$.
- Scenarios, $s = 1, \ldots, 8$.
- Assets, $i = 1, 2$.
- Bid-ask spread, $b = 0.03$.
- Initial wealth in cash, $W_0 = 100$.
- Initial portfolio composition $\mathbf{x}_0^\top = [10 \quad 10]$.

Agenda

- We overview the GAMS language through three simple optimization problems:

1. Deterministic program of a simple refinery model
2. Simple asset-liability (ALM) model
3. Four-stages ALM stochastic programming model

4-stages ALM Model

$$\min_{\mathbf{x}} \Gamma = \mathbf{c}^\top \mathbf{x} \qquad (1)$$

subject to

$$\mathbf{b}^\top \mathbf{x} \leq q \qquad (2)$$

$$\Pi^\top \mathbf{x} \geq \mathbf{d} \qquad (3)$$

Γ Total costs of crude oil consumption

$\mathbf{x}$ Unit of crude oil consumed in the production

$\mathbf{c}$ Costs of crude oil ($ per barrel)

$\mathbf{b}$ Marginal production capacity

q Total refinery capacity

Π Unit of crude oil per unit of product

$\mathbf{d}$ Total demand of each product

Simple Refinery Model

Input Data

- Type of crude oil (light and heavy), $j = 1, 2$.
- Products of refinery (e.g. gasoline and diesel), $k = 1, 2$.
- Total refinery capacity, $q = 15000$.
- Product's demand, $\mathbf{d}^\top = [13200 \;\; 8000]$.
- Marginal production capacity, $\mathbf{b}^\top = [55 \;\; 55]$.
- Costs of crude oil ($ per barrel), $\mathbf{c}^\top = [42 \;\; 22]$.
- Unit of crude oil per unit of product, $\Pi = \begin{bmatrix} 200 & 100 \\ 60 & 50 \end{bmatrix}$.