

## **Vaccine Manufacturing Modelling Webinar**

**Developing Countries Vaccine Manufacturers Network (DCVMN)** 

**13 December 2018** 

Zoltan Kis, Maria Papathanasiou, Cleo Kontoravdi and Nilay Shah Centre for Process Systems Engineering Chemical Engineering





- Vaccine Manufacturing Hub
- Work Stream 1 (WS1): Improving existing processes and platforms
- Work Stream 2 (WS2): Design, development and implementation of new platform technologies
- Modelling tools:
  - gPROMS (unit operation level);
  - SuperPro Designer (process flow sheet level);
  - GAMS (manufacturing and supply chain level).





- Work Stream 1 (WS1): Improving existing processes and platforms
- Work Stream 2 (WS2): Design, development and implementation of new platform technologies
- Modelling tools:
  - gPROMS (unit operation level);
  - SuperPro Designer (process flow sheet level);
  - GAMS (supply chain level).

# WS1. Integrating life science and engineering to improve existing platforms

- LMIC partners have platforms for the production of vaccine formulations suitable for LMIC environment
- WS1 focuses on how best to exploit and further develop these to
  - enable mass vaccination campaigns
  - deal effectively with new outbreaks
- Objectives:
  - operational efficiency for cost reduction
  - rapid response of existing assets
  - end-to-end system design
- How? Whole process analysis and optimisation to address bottlenecks
  - In Life Sciences: host cell system or vector optimisation for improved productivity/quality
  - In Engineering: downstream separations, formulation and packaging

### WS1. Methodology

- Current capabilities
  - 90% of the LMIC vaccine production is attenuated and inactivated bacterial/virus vaccines
  - 10% mostly conjugates
- We have established a computational platform for modelling and optimising vaccine manufacturing processes to reduce costs
  - Use this to create parallel models that describe existing capabilities in LMIC partners and possible alternatives
- Apply whole process optimisation, system design and process intensification to improve operational flexibility and efficiency
- Process intensification has great promise for cost reduction and improvement of responsiveness in vaccine manufacturing

### WS1. Methodology

- Model, simulate and optimize the manufacturing and delivery processes at:
  - I. unit operation level using gPROMS;
  - II. process flow sheet level using SuperPro Designer,
  - III. supply chain level using GAMS
- Perform stochastic sensitivity analysis to determine the input variables that have the highest potential for cost reduction, then further minimize costs by adjusting the high-impact variables.
- Enhance bioprocessing by:
  - a) process intensification continuous processing strategies;
  - b) de-bottlenecking;
  - c) "process telescoping" combining several unit operations into one (e.g. charge and size based separation in one unit operation).
- Integration with experimental work, iterative



#### **WS1. Modular Design**



- Advantages of a modular design:
  - Plug & Play
  - Standardization
  - Flexibility
  - Streamlined troubleshooting
  - Easy of training operators

# WS1. Rapid prototyping of novel downstream separation process concepts

- Explore two purification concepts for whole virus/bacteria, subunit and proteins:
  - "process telescoping" whereby several unit operations are combined into one (e.g. expanded bed affinity adsorption combining solids removal, capture and primary purification)
  - continuous operation (e.g. moving to continuous chromatography using simulated moving bed technology).
- Our key activities will involve high throughput experiments, models and big data analytics.
- Deliverable: Demonstration of new vaccine separation design concepts at lab scale



### WS1. Key drivers for continuous manufacturing

- Manufacturing capacity
- Reduced footprint, labour costs and CapEx
- Flexibility
- Speed to market
- Improved quality through the application of QbD & PAT





### **WS1. Challenges**

#### Design Control Design and control interactions • Complex operation profiles Complicated process models Bioreactor design • Unavailable measurements Downstream setup configuration • **Regulatory Bodies Operation** Optimal setup configuration Enhanced process understanding • Monitoring Feasible operation ٠ • **Global** control Task coordination (scheduling) ٠ •

### WS1. Requirements for a smooth transition

- ✓ Quality by Design (QbD)
- ✓ Thorough understanding
- ✓ Identification of risks/bottlenecks
- ✓ Process monitoring
- ✓ Global process control



.... **minimizing** experimental time & cost....

#### Imperial College London WS1. Development of computational models for whole systems analysis

- Scaled up designs will be used to explore supply chain configuration:
  - centralised vs decentralised
  - shipment of bulk or fully-filled vaccines to clinics/local fill-finish plants
- What is a Supply Chain?
  - The alignment of firms that bring products or services to market
  - Linked by counter-current flow of material and information



#### **Supply Chain Management**

"The systemic, strategic coordination of the traditional business functions and the tactics across these business functions within a particular company and across businesses within the supply chain, for the purposes of improving the long- term performance of the individual companies and the supply chain as a whole."

### WS1. Results

- State of the art conventional bacterial, viral and recombinant vaccine manufacturing processes have been reviewed.
- Calculated the batch volume for producing 1500 doses of the Hand, foot, and mouth disease (HFMD) vaccine in yeast at Dalian Hissen BioPharm Co., Ltd. (China).

### WS1. Ongoing case studies

- Analysing and improving vaccine manufacturing processes at MSD Wellcome TrustHilleman Laboratories Pvt. Ltd. (Hilleman Labs, India) for Recombinant cholera toxin B subunit and whole cell cholera vaccine production.
- Developing a model of the Vero cell based whole viral vaccine production and perform sensitivity analysis to quantify the relationships between CPPs and CQAs in support of QbD of this family of vaccine manufacturing processes. In collaboration with VaBiotech (Vietnam).
- We are engaging with additional vaccine developers and manufacturers in developing countries to simulate and optimize their processes (e.g. DCVMN meeting and webinars).





- Work Stream 1 (WS1): Improving existing processes and platforms
- Work Stream 2 (WS2): Design, development and implementation of new platform technologies
- Modelling tools:
  - gPROMS (unit operation level);
  - SuperPro Designer (process flow sheet level);
  - GAMS (supply chain level).

### WS2. The 4 emerging vaccine platform technologies

#### **RNA** vaccines

- ~10 kb self-amplifying RNA
- regular mRNA
- produced by *in vitro* transcription

#### **ADDomer vaccines**

- <u>ad</u>enovirus <u>do</u>decahedron derived multi<u>mer</u>
- 3.6 MDa VLP, 360 configurable epitopes, different epitope types
- insect cell-baculovirus expression

#### Humanized yeast vaccines

- glycoengineered yeast
- high-yield yeast expression of recombinant proteins

#### **GMMA** vaccines

- <u>Generalized Modules for Membrane</u>
   <u>Antigen, outer membrane vesicles</u>,
   20-60 nm diameters
- configurable membrane proteins, bacterial expression

### WS2. Platform evaluation and improvement metrics

Speed	~100,000 vaccine doses, weeks after threat antigen identification			
Cost	low cost, below 1 \$/dose			
Flexibility	on-demand production of a wide range of vaccine types (viral and bacterial)			
Technological complexity	low technological complexity for implementation in developing countries			
Technology readiness	mature technologies with licensed products & established manufacturing processes			
Ease of scale-up or -out	highly scalable upstream and downstream processes			
Thermo-stability of product	vaccines stable at 40°C for at least 6 months			



### **WS2. Workflow**



### WS2. Methodology

- To overcome data scarcity and uncertainty:
  - i. Calculate parameters on a first-principle, bottom-up basis;
  - ii. Estimate parameters based on similar processes and products by surveying the scientific literature and patent databases, and by interviewing experts;
  - iii. Model, simulate and optimize the manufacturing processes at unit operation level (using gPROMS) and process flow sheet level (using SuperPro Designer, Aspen Batch Process Developer), at supply chain level (using GAMS)
- Perform stochastic sensitivity analysis to determine the input variables that have the highest potential for cost reduction, then further minimize costs by adjusting the high-impact variables.
- Enhance bioprocessing by: (a) evaluating process intensification continuous processing strategies; (b) de-bottlenecking; (c) "process telescoping" combining several unit operations into one

### WS2. RNA platform results



- ~10 kb self-amplifying RNA
- produced by *in vitro* transcription
- cell-free product
- Co-transcriptional 5' capping, ARCA

Cost per dose at 20 µg/dose: 0.72 USD/dose



## Imperial College London WS2. ADDomer platform results

- <u>adenovirus</u> <u>do</u>decahedron derived multi<u>mer</u>
- 3.6 MDa VLP, 360 configurable epitopes, different epitope types
- insect cell-baculovirus expression
- intracellular





### WS2. Humanized yeast platform results

• glycoengineered yeast

**Imperial College** 

London

- high-yield yeast expression of recombinant proteins
- extracellular





### **WS2. GMMA platform results**

• <u>Generalized Modules for</u> <u>Membrane Antigen</u>, outer membrane vesicles, 20-60 nm diameters

• configurable membrane proteins, bacterial expression

• extracellular





### WS2. Feasibility and risk assessment results

Table 1. Feasibility and risk assessment of the 4 emerging platform technologies

	Platform <sup>a)</sup>	Yeast	ADDomer	GMMA	RNA
Metric		platform	platform	platform	platform
1	Technology readiness	2	3	5	4
2	Technological complexity	3	1	5	2
3	Ease of scale-up and -out	4	2	5	3
4	Flexibility <sup>b)</sup>	3	3	2	4
5	Thermo-stability of product	3	5	3	2
6	Speed of response	1	4	2	5
Sum: overall feasibility and risk estimate <sup>c)</sup>		16	18	22	20

<sup>a)</sup> Yeast platform - Humanised, high-yield yeast platform for recombinant vaccine manufacturing; ADDomer platform - Insect cell-baculovirus platform for recombinant vaccine manufacturing; GMMA platform - Outer membrane vesicle vaccines, Generalized Modules for Membrane Antigen vaccine manufacturing; RNA platform - RNA vaccine manufacturing.

<sup>b)</sup> Universal applicability for the manufacturing of a wide range of vaccines.

<sup>c)</sup> The overall feasibility and risk estimate was calculated by summing up the values for each metric per technology.

Zoltán Kis, Robin Shattock, Nilay Shah, Cleo Kontoravdi. Emerging technologies for low-cost, rapid vaccine manufacture. Biotechnology Journal. Accepted. Oct 2018.

### **Process Systems Engineering – lifecycle view**





#### WS2. Outcomes

Key outcomes from this workstream include:

- A screening methodology which can identify which platform is best suited to a particular vaccine
- Conceptual and detailed demonstration industrial process designs/blueprints which build on lab data
- Actual physical demonstrations of new manufacturing concepts with emerging data
- Application of these to demonstrate the benefits of novel platforms and approaches as an evidence base for regulators, healthcare providers and manufacturers.

### WS2. Future tasks

Task 2.3: Application of detailed demonstration design methodology to up to 6 shortlisted demonstration process concepts, leading to:

(a) detailed designs and

(b) more accurate KPIs including cost,

which will be used to select the demonstration projects for physical deployment.

Task 2.4: Development of physical demonstrators - up to 6 scale appropriate physical demonstrators used to:

- advance the knowledge around industrialisation of the new platforms,
- evaluate actual performance in an industrial setting,
- feed back information to the model-based design activity to support analysis and design of full commercial scale processes.





- Work Stream 1 (WS1): Improving existing processes and platforms
- Work Stream 2 (WS2): Design, development and implementation of new platform technologies



#### • Modelling tools:

- gPROMS (unit operation level);
- SuperPro Designer (process flow sheet level);
- GAMS (supply chain level).

### Methodology

- We have established a computational platform for modelling and optimising vaccine manufacturing processes to reduce costs
  - Use this to create parallel models that describe existing capabilities in LMIC partners and possible alternatives
- Apply whole process optimisation, system design and process intensification to improve operational flexibility and efficiency
- Process intensification has great promise for cost reduction and improvement of responsiveness in vaccine manufacturing



### Why model?



Innovation, risk and modelling – models and data not models or data

## Competitive advantage



# Application of integrated approach for improving primary production

- Employ high-level models of traditional upstream systems (bacteria, yeast and animal cells) and our emerging platform
- Optimise using process mapping, bottleneck identification and process intensification, building on work in biologics manufacturing.
- Also identify raw materials and suitable alternatives available locally and/or at lower cost.
- Deliverable: Demonstration of benefits of integrated approach on primary production systems

#### **Development of computational models for whole systems analysis**

- Multi-scale modelling of biological processes through to unit operation and whole value chains will be used for system analysis, design and manufacturing operation optimisation
- How do parameters characterising single unit performance e.g. titre, purity, recovery, formulation recipe influence whole system metrics e.g. cost per dose, lead times?
  - Identify priority areas of study

### **Modelling for process optimisation**

Following Quality by Design principle for increasing process understanding





#### **Development of reliable process models**











#### In silico optimisation

#### Cell Culture Dynamics




9.00 8.80

8.60

Cell culture period (Day)



### Whole process analysis: intracellular product



### Initial process design and dynamic optimisation



*Figure 3.* Optimal fermenter profiles, for problem 1 (Table 1).---- P; -.-- S; --- X;  $---- \mu$ .

**Imperial College** 

London

Figure 5. Stress and cell strength distribution in homogenizer, problem  $1, --- D; \blacksquare f-D; \bullet f-s.$ 

### Simultaneous design of key variables

ruole 2. Optimar design for b	Table 5. Opt		
Fermenter	$\begin{array}{c} S_0 \\ X_0 \\ V (= V_0) \\ N \\ I_{\text{form}} \end{array}$	101.04 5 800 1 11.888	Fermenter
Cell harvesting centrifuge	F <sub>feed</sub> w t <sub>heant</sub>	110 10000 7.273	Cell harvesti
Homogenizer	F P t <sub>hom</sub>	60 55 13.333	Homogenize
Centrifuge 1	$F_{\text{feed}} \\ \omega \\ t_{\text{cent 1}}$	80 7234.3 10	Centrifuge I
Precipitation stage 1	$rac{M_{ m salt,in}}{M_{ m salt}}$	0 32000	Precipitation
Centrifuge 2	$F_{\text{feed}}$	80 6500 9.5	Centrifuge 2
Precipitation stage 2	M <sub>salt,in</sub> M <sub>salt</sub>	32000 28000	Precipitation
Centrifuge 3	$F_{\text{feed}}$ $\omega$ $t_{\text{cent 3}}$	75 9500 10.397	Centrifuge 3
Overall cycle time	t <sub>exe</sub>	13.333	Overall cycle
Process performance	profit purity recovery	48311 0.1299 0.7585	Process perfo

Table 2. Optimal design for batch fermentation, maximizing profit, Table 3. Optimal design for batch fermentation, maximizing purity.

Fermenter	$V \stackrel{S_0}{\underset{M}{\overset{X_0}{(=}}} V_0)$	107.61 5 800 1 11.655
Cell harvesting centrifuge	$F_{\text{faed}} = \frac{\omega}{t_{\text{hcent}}}$	115.85 9042.3 6.905
Homogenizer	$F P t_{hom}$	60 80 13.333
Centrifuge 1	$F_{ ext{food}} \ _{\omega}^{\omega} t_{ ext{cont 1}}$	60 2274.8 13.333
Precipitation stage 1	$rac{M_{ m salt,in}}{M_{ m salt}}$	0 31114
Centrifuge 2	$F_{\text{faad}} \\ \substack{\omega \\ t_{\text{cent}2}}$	70.376 7391.8 10.799
Precipitation stage 2	$M_{ m salt,in} \ M_{ m salt}$	31114 18945
Centrifuge 3	$F_{\text{faed}} \\ {}^{\omega} \\ t_{\text{cent}3}$	60 8000 12.033
Overall cycle time	t <sub>cyc</sub>	15.946
Process performance	profit purity recovery	12505 0.1395 0.7191

### From design to operations



Figure 1. State task network for the example problem.

State	Tank capacity (l)		
feed and product states	not sized		
broth	800		
cells	24.48		
dcells	740		
int l	125		
liquid 1	684		
ppt1	760		
liquid2	684		
ppt2	652.34		

Table 1. Optimal storage tank capacities.



Figure 4. Cyclic schedule for one batch, as above, showing three batches.



### **Debottlenecking of influenza vaccine process**





#### **Debottlenecking model**



#### Table 2. Comparisons of different methods debottlenecking strategies.

Im	pe	rial	Col	lege
Lo	nd	on		

. . . .

**Debottlenecking** 

Case Batches obtained

(% ch	ange)	Bottleneck	Customer service level		
A	67	Incubation	93%		
В	93 (39%)	Incubation	100%		
C	98 (46%)	Inactivation	100%		
D	98 (46%)	Incubation	100%		



### **Incorporation of uncertainty**



# The aid of computational tools



#### Imperial College London Development of computational models for whole systems analysis

- Multi-scale modelling of biological processes through to unit operation and whole value chains will be used for system analysis, design and manufacturing operation optimisation
- How do parameters characterising single unit performance e.g. titre, purity, recovery, formulation recipe influence whole system metrics e.g. cost per dose, lead times?
  - Identify priority areas of study





- Work Stream 1 (WS1): Improving existing processes and platforms  $\checkmark$
- Work Stream 2 (WS2): Design, development and implementation of new platform technologies
- Modelling tools:
  - gPROMS (unit operation level);
  - SuperPro Designer (process flow sheet level);
  - GAMS (supply chain level).



## **Introduction to gPROMS**

**Describing new processes and unit operations** 

**Dynamic process modelling** 

- Seamless modelling capabilities (no manual discretization required)
- Other functionalities: dynamic optimisation, parameter estimation based on experimental data, experiments to be designed





### **Main navigation window**



## **Declaration of variables & bounds**

Hie Edit Vew tritty Activities Tools Window Help  Variable Type: (Bioreactor Concertration Concertra	gPROMS ModelBuilder 5.0.1								- 0
Conficient     C	File Edit View Entity Activities Tools Window Help								
Image: Second	- < 8 € € 8 k k k k k k k k k k k k k k k k								
Image: Subject of the system       Image: Subject of the system       Concentration       Unit of the system       Concentration       Upper bound       Upper bound       Upper bound       Image: Subject of the system		○ Variable Types (Bio	reactor)					c	- 🗆 💌
• coefficient       "mg/mg"       3.00000       0.00000       100000         • concentration       "mg/"       2.0000       0.00000       100000         • concentration       "mg/"       2.0000       0.00000       100000         • flow_rate       "U/day"       2.0000       0.00000       100000         • mass       "mg"       0.50000       0.00000       100000         • no type       "mg"       0.50000       0.00000       100000         • rate       "U/day"       2.00000       0.00000       100000         • ass       "mg"       0.50000       0.00000       100000         • rate       "U/day"       2.00000       0.00000       100000         • ata       "U/day"       0.020000       0.00000       100000         • ata       "U/day"       0.020000       0.00000       100000         • potesses       • parameter Estimations       • parameter Estimations       • optimisations       • optimisations         • Optimisations       • Optimisations       • prostise       • mg/mg"day"       • uptake_rate	Fioreactor      Fioreactor	Name	Quantity type	Unit	Delta	Default value	Lower bound	Upper bound	New
• concentration       "mg/L"       2.0000       0.00000       100.000         • concentration       "mg/L"       2.0000       0.00000       100.000       100.000         • flow, rate       "growth_rate       "1/day"       2.0000       0.00000       100.000       100.000         • mass       • mg/L"       2.0000       0.00000       100.000       100.000       100.000         • mass       • mg/L"       2.0000       0.00000       100.000       100.000       100.000         • mass       • mg/L"       2.0000       0.00000       100.000       100.000       100.000         • mass       "mg/L"       2.0000       0.00000       100.000       100.000       100.000         • rate       • "1/day"       2.0000       0.00000       100.000       100.000       100.000         • sat_const       • mg/Tmg/T       0.020000       0.00000       10.0000       10.0000       10.0000       10.0000         • uptake_rate       * mg/Tmg/Tay"       0.100000       0.00000       10.0000       10.0000       10.0000       10.0000       10.0000       10.0000       10.0000       10.0000       10.0000       10.0000       10.0000       10.0000       10.0000       10.0000	coefficient	coefficient		"mg/mg"		3.00000	0.00000	10.0000	Edit
Occor_error         Iow rate         Iow rate <thiow rate<="" th="">         Iow rate         <thiow rate<="" th=""></thiow></thiow>	concentration	concentration		"mg/L"		2.00000	0.00000	1000.00	
Image: Constant Service	conc_error	conc_error				1.00000	-100.000	100.000	Delete
• growth_rate         • mass         • no_type         • no_type         • rate         • fate	flow_rate	flow_rate		"L/day"		2.00000	0.00000	200.000	
• mass       "mg"       0.50000       0.00000       100.000         • no_type       3.0000       0.00000       100.000         • sat_const       "mg/"       0.020000       0.00000       100.000         • uptake_rate       "mg/"       0.020000       0.00000       100.000         • motype       "mg/"       0.020000       0.00000       100.000         • uptake_rate       "mg/"       0.020000       0.00000       10.0000         • motype       "mg/"       0.100000       0.00000       10.0000         • uptake_rate       "mg/mg"day"       0.100000       0.00000       10.0000         • processes       • Experiments       • Processes       • Uptake_rate       "mg/mg"day"       0.100000       0.00000       10.0000         • prometer fstimations       • Optimisations       • Optimisations       • Uptake_rate       • Uptake_rate       • Uptake_rate	growth_rate	growth_rate		"1/day"		2.00000	0.00000	10.0000	
<ul> <li>no_type</li> <li>rate</li> <li>sat_const</li> <li>uptake_rate</li> <li>Models</li> <li>Tasks</li> <li>Processes</li> <li>Experiments</li> <li>Optimisations</li> <li>Optimisations</li> </ul>	🗢 mass	mass		"mg"		0.500000	0.00000	10000.0	
<ul> <li>rate</li> <li>"1/day"</li> <li>2.0000</li> <li>0.0000</li> <li>100.00</li> <li>sat_const</li> <li>"mg/l"</li> <li>0.020000</li> <li>0.0000</li> <li>10.000</li> <li>10.000<td>no_type</td><td>no_type</td><td></td><td></td><td></td><td>3.00000</td><td>0.00000</td><td>100.000</td><td></td></li></ul>	no_type	no_type				3.00000	0.00000	100.000	
Image: Sat_const       Image: Im	🗢 rate	rate		"1/day"		2.00000	0.00000	1000.00	
uptake_rate     Models   Tasks   Processes   Experiments   Parameter Estimations   Optimisations     Optimisations     Tore definitions	sat_const	sat_const		"mg/l"		0.0200000	0.00000	10.0000	
<ul> <li>Models</li> <li>Tasks</li> <li>Processes</li> <li>Experiments</li> <li>Optimisations</li> <li>Optimisations</li> </ul>	🗢 uptake_rate	uptake_rate		"mg/mg*day"		0.100000	0.00000	10.0000	
<ul> <li>Tasks</li> <li>Processes</li> <li>Experiments</li> <li>Parameter Estimations</li> <li>Optimisations</li> </ul>	🗄 🚞 Models								
<ul> <li>Processes</li> <li>Experiments</li> <li>Parameter Estimations</li> <li>Optimisations</li> </ul>	🗄 💼 Tasks								
Experiments  Parameter Estimations  Optimisations  Ture definition:  Properties	E Processes								
Parameter Estimations Optimisations  Comparison	🗄 🚞 Experiments								
B Optimisations	🗄 🚞 Parameter Estimations								
	I Detimisations								
Ture definitions Properties									
Type definitions Properties									
Type definitions Properties									
Type definitions Properties									
Projects Palette	Projects Palette	Type definitions Pro	perties						

### **Model entity**

gPROMS ModelBuilder 5.0.1					- 0 ×
File Edit View Entity Activities Tools \	Window	v Help			
	50		•		
	D Ph	notobioreactor_CSTR (I	Bioreactor)		
	1	PARAMETER			Declaration of:
T Variable Types	2	no_species	AS INTEGER	# Number of Species	
	3	Vr	AS REAL	# CSTR Volume	
Photobioreactor CSTP	5	bita	AS REAL	# Fitty Acid Synthesis Coefficient	
	6	gama	AS REAL	# Fatty Acid Mobilization Coefficient	Devenetere
	7	miu bar	AS REAL	# Theoretical Maximum Growth Rate	- Parameters
	8	Qo	AS REAL	# Minimal Nitrogen Quota	N7 * 1 I
🗄 🔜 lasks	9	cm	AS REAL	# Maximal Uptake Rate	- Variables
Processes	10	Ks	AS REAL	# Half Saturation Constant	
🖽 🚞 Experiments	11	roe	AS REAL		- Equations
🗄 🚞 Parameter Estimations	12				Equations
🖽 🚞 Optimisations	13	VARIABLE			
· · · · · · · · · · · · · · · · · · ·	14	Fi	AS flow_rate		
	15	Frec	AS flow_rate		
	16	Fr	AS flow_rate		
	10	FSET	AS flow_rate		
	10	Fout	AS IIOw_rate		
	20	si	AS concentration		
	21	sr	AS concentration		
	22	rec	AS ARRAY (no spec:	ies) OF concentration	
	23	r	AS ARRAY (no spec:	ies) OF concentration	
	24	ex	AS ARRAY (no spec:	ies) OF concentration	
	25	g	AS concentration		
	26	qn	AS no_type		
	27	ql	AS no_type		
	28	qf	AS no_type		
	29	miu	AS rate		
	30	C	AS rate		
-	31	1.1			· · · · · · · · · · · · · · · · · · ·
< >>		1.1 1N5			
Projects Palette	Inte	erface Interface langua	age Topology gPROMS la	anguage Properties	li -

### **Model entity**

gPROMS ModelBuilder 5.0.1 File Edit View Entity Activities Tools V	Window Help	– 0 X
<ul> <li>Bioreactor</li> <li>Variable Types</li> <li>Models</li> <li>Photobioreactor_CSTR</li> <li>Photobioreactor_CSTR</li> <li>Photobioreactor_CSTR</li> <li>Photobioreactor_CSTR</li> <li>Processes</li> <li>Experiments</li> <li>Parameter Estimations</li> <li>Optimisations</li> </ul>	Photobioreactor_CSTR (Bioreactor) 2 EQUATION 3 # Total Mass Balance around Mixer 4 Fi + Frec = Fr; 5 # Mixer Mass Balances 6 Fi*si = Fr*sr; 7 FOR i := 1 TO no_species DO 8 Frec*rec(i) = Fr*r(i); 9 END 40 41 # Total Mass Balance around Reactor 42 $0 = Fr - Fset;$ 43 # Reactor Mass Balances 44 \$s = Fr*sr/Vr - Fset*s/Vr - c*ex(1); 45 \$ex(1) = Fr*r(1)/Vr - Fset*ex(2)/Vr + miu*ex(1); 46 \$ex(2) = Fr*c(2)/Vr - Fset*ex(2)/Vr + c*ex(1); 47 \$ex(3) = Fr*r(3)/Vr - Fset*ex(3)/Vr + bita*ex(2)*miu - gama*c*ex(1); 48 \$ex(4) = Fr*r(4)/Vr - Fset*ex(4)/Vr + (alpha + gama)*c*ex(1); 49 $g = ex(1) - ex(3) - ex(4);$ 50 51 # Total Mass Balance around Settling Tank 52 For i := 1 TO no_species DO 53 # Settling Tank Mass Balances 54 # Fset*s = Fout*frec(i); 55 FOR i := 1 TO no_species DO 66 Fset*ex(i) = Frec*rec(i); 57 END 58 59 # Quota Definitions 60 $qn = ex(2)/ex(1);$ 61 $ql = ex(3)/ex(1);$	<ul> <li>Description of equations</li> <li>Possible to include: <ul> <li>ODEs</li> <li>DAEs</li> <li>PDAEs</li> </ul> </li> </ul>
Projects Palette	1:1     INS       Interface     Interface language       Topology     gPROMS language	

### **Process entity**

gPROMS ModelBuilder 5.0.1						- 0 ×	
File Edit View Entity Activities Tools V	Vindo	w Help					
🍅 📽 🔲 🗐 🖉 🗛 🐚 👘	50		•				
	► C	yclic_CSTR (Bioreacto	or)				
	1	UNIT				^	
Variable Types	2	Reactor AS Ph	otobioreactor	_CSTR			
E Models	4	SET			<ul> <li>Link process to</li> </ul>		
Photobioreactor CSTR	5	WITHIN Reacto	or DO				
Photobioreactor CSTR	6	no_species	:= 4;		model		
Photobioreactor CSTR	7	Vr	:= 5;	# (L)	IIIUUEI		
I Tasks	8	alpha	:= 2.6;	# mg[C]/mg[N]	<ul> <li>Assign values to</li> </ul>		
	9	bita	:= 4.8;	# mg[C]/mg[N]	<ul> <li>Assign values to.</li> </ul>		
Curlia CCTD	10	gama	:= 3.0;	# mg[C]/mg[N]			
	11	miu_bar	:= 2.11;	# 1/day	-Parameters		
Cyclic_CSTR_INITIAL_VA	12	Qo	:= 0.05;	# mg[N]/mg[C]			
Cyclic_CSTR_INITIAL_VA	13	Cm	:= 0.095;	# mg[N]/mg[C]/day	_\/ariables		
Cyclic_CSTR_OPT	14	KS NOO	:= 0.018;	+ mg[N]/L	Variabies		
Cyclic_CSTR_PARAM_ES	15	END	0.5,				
Cyclic_CSTR_TESTING_T	17	END					
Cyclic CSTR TESTING T	18	ASSIGN					
Cvclic CSTR TESTING T	19	WITHIN Reacto	r DO				
Experiments	20	Frec	:= 10;	# L/day			
Paramotor Estimations	21	Fi	:= 1;	# L/day			
	22	si	:= 5;	# mg[N]/L			
Opumisauons	23						
	24	END # Within					
	25						
	26	INITIAL					
	27	WITHIN Reacto	or DO				
	28	ex(1)	= 10;	# mg[C]/L			
	29	qn	= 0.06;	# mg[N]/mg[C]			
	30	ql	= 0.12;	# mg[C]/mg[C]			
	31	qf	= 0.40;	# mg[C]/mg[C]			1
< >>		1:1 INS					L
Projects Palette	Sc	hedule Solution para	ameters gPROMS	language Properties		Demokra 2010	
					11.	December 2018	-
					Tue	Suuy	

gPROMS ModelBuilder 5.0.1

File Edit View Entity Activities Tools Window Help

### **Parameter estimation**

• Performed experiments

D

×

- Experimental data
- Experimental error

「 「 「 「 」 「 」 「 」 「 」 「 」 「 」 「 」 「 」 「 」 」 」 「 」 」 」 」 の つ い つ つ い つ つ い つ い つ い つ い つ い つ い つ い つ い つ い つ い つ い つ い つ い つ つ つ つ つ つ つ つ つ つ つ つ つ	🔎 ເ 🕤 🗟 🕨 -							
	🕌 Experiment_1 (Bioreacto	r)						d ×
Bioreactor	Time							
🗄 🛅 Variable Types	Variable Name	Reactor.ex(1)	Reactor.qg	Reactor.ql	Reactor.qn	Reactor.s	Add new variable	^
🖽 🧮 Models	Sensor	Reactor.ex(1)	Reactor.qg ~	Reactor.ql ~	Reactor.qn ~	Reactor.s ~		
🖽 🚞 Tasks	Variance model	CONSTANT_RELATIVE_VARIAN	CONSTANT_RELATIVE_VARIAN	CONSTANT_RELATIVE_VARIAN	CONSTANT_RELATIVE_VARIAN	CONSTANT_RELATIVE_VARIAN		
🗄 🚞 Processes		"mg/L" ~	~	~	~	"mg/L" 🗠 📉		
Experiments	0.0079000	0				1.53790		
🖃 🛅 performed	0.13290	0				1.52610		
Experiment_1	0.82920	0 27.4431	0.277800		0.0869000			
Experiment_2	0.84310	0		0.166900				
Parameter Estimations	1.9155	0				1.36750		
Param_esum	1.9472	0 26.4392	0.314300		0.0918000			
Param Estim 20181211 113033	1.9542	D		0.187800				
Criginal Entities	2.0002	0				1.21600		
H Variable Types	2.0835	0				0.983200		
I To Models	2.1669	0				0.730500		
🗄 📷 Processes	2.2502	0				0.441600		
🗄 📷 Experiments	2.3335	0				0.0932000		
🕀 💼 Parameter Estimations	2.6669	0				0.0129000		
🖃 📠 Results	3.2502	0				0.0102000		
Param_Estim.gradient	3.3335	D				0.0112000		
Param_Estim.hessian	3.5002	D				0.0255000		
Param_Estim.out	3.8458	0				0.127700		
Param_Estim.params	3.9291	D				0.0155000		
Param_Estim.stat-mr		p			1	0.0000000	-	×
Param_Estim			Select measured variables	. Delete row D	elete column Transp	oose		
Projects Palette	General Controls Measu	ured data gPROMS langua	ge Properties					

### **Parameter estimation**

Simulation vs experimental data

• Here: estimation of 1 parameter

gPROMS ModelBuilder 5.0.1
 File Edit View Activities Tools Window Help

늘 📲 🖶 🛢 🖡 👗 🐂 🛍 🖉 🖉 🗑 🖗 📲 😫



П



### **Open loop model simulation**



đ X

#### **Dynamic optimisation**

đ gPROMS ModelBuilder 5.0.1 X File Edit View Entity Activities Tools Window Help OPTIMIZATION\_Time\_Varying\_F (Bioreactor) - 6 × Cyclic CSTR OPT 🗏 🛅 Bioreactor Process V 🗄 🚞 Variable Types Objective function Reactor.obj Select... 🗄 🚞 Models Type of optimisation Dynamic  $\sim$  Maximise 
 Minimise 🗄 🚞 Tasks 🗄 🚞 Processes Time unit 🗄 🚞 Experiments Time horizon 🗄 🚞 Parameter Estimations 😑 🛅 Optimisations 1.00000 ✓ Fixed Guess OPTIMIZATION\_Constant\_F Lower bound 1.00000 OPTIMIZATION Time Varying F OPTIMIZATION\_Time\_Varying\_F\_T1 Upper bound 1.00000 OPTIMIZATION\_Time\_Varying\_F\_T2 OPTIMIZATION\_Time\_Varying\_F\_T3 Control intervals To Duration Fixed? Lower bound Upper bound From ~ 0.00000 0.100000 0.100000 0.100000 0.100000 ~ 0.100000 0.200000 0.100000 0.100000 0.100000 ~ 0.300000 0.100000 0.100000 0.200000 0.100000  $\checkmark$ 0.400000 0.100000 0.100000 0.100000 0.300000 ~ 0.400000 0.500000 0.100000 0.100000 0.100000  $\checkmark$ 0.500000 0.600000 0.100000 0.100000 0.100000  $\checkmark$ 0.600000 0.700000 0.100000 0.100000 0.100000  $\checkmark$ 0.700000 0.800000 0.100000 0.100000 0.100000  $\checkmark$ 0.800000 0.900000 0.100000 0.100000 0.100000 ~ 1.00000 0.900000 0.100000 0.100000 0.100000 Duplicate General Controls Constraints gPROMS language Properties Projects Palette

#### Declaration of process profile



#### **Dynamic optimisation**

#### Declaration of control variables (manipulated)

đ

 $\times$ 

File Edit View Entity Activities Tools Window Help

**B** gPROMS ModelBuilder 5.0.1

	© OPTIMIZATION_Time_Varying_F (Bioreactor)					
	Variable		Control type	Allowable values		Select
	Reactor.ex1_0		Time-invariant	Continuous	^	
	Reactor.Fi		Piecewise-constant	Continuous		Delete
Tasks	Reactor.qf_0		Time-invariant	Continuous		
	Reactor.ql_0		Time-invariant	Continuous		
Experiments	Reactor.qn_0		Time-invariant	Continuous	~	
Parameter Estimations	Control type	Allowable value	es			
🗆 🛅 Optimisations	Time-invariant	Continuous				
OPTIMIZATION_Constant_F	O Piecewise-constant	Binary		Integer		
	O Piecewise-linear		1	O Special Ordered Set 1		
OPTIMIZATION_Time_Varying_F_T2     OPTIMIZATION_Time_Varying_F_T3	Unit "mg/L"  Bounds Initial guess 13.0950 Lower bound 1,00000					
D Delatta	Upper bound 100.000					
Projects Palette	Controls Constraints grkOws language Properties				1	

#### **Dynamic optimisation**

#### Declaration of constraints

þ

×

File Edit View Entity Activities Tools Window Help

gPROMS ModelBuilder 5.0.1

OPTIMIZATION Time Varying F (Bioreactor)	E X
Equality end-point constraints	
Constrained variable Types Unit Constrained value	Select
Add new	Delete
Tasks	Delete
H Processes	
III Experiments	
🗄 🚞 Parameter Estimations	
E Continuisations	
OPTIMIZATION_Constant_F	
OPTIMIZATION_Time_Varying_F	
OPTIMIZATION_Time_Varying_F_T1 Constrained variable Unit Lower bound Upper bound	Select
OPTIMIZATION_Time_Varying_F_T2         Reactor.dex1         -0.000100000         0.000100000	Delete
Image: Control of the sector decomposition of the sector decomp	
Reactor.dex3 -0.000100000 0.000100000	
Reactor.dex4 -0.000100000 0.000100000	
Reactor.dex5 -0.000100000 0.000100000	
Reactor.ex(1)         "mg/L"         0.00000         100.000	
Interior-point constraints	
Constrained variable Unit Varying? At start of inter Lower bound Upper bound	Select
Add new	0.14
	Delete
Projects Palette General Controls Constraints gPROMS language Properties	

gPROMS ModelBuilder 5.0.1

## **Dynamic optimisation – Results**

#### Suggested optimal values following 5 iterations

- 0 ×





### **Getting Help**

• In gPROMS click on "Help" menu then click on "Documentation".



• PSE Webinars:

https://www.psenterprise.com/events/webinars





- Work Stream 1 (WS1): Improving existing processes and platforms
- Work Stream 2 (WS2): Design, development and implementation of new platform technologies



- Modelling tools:
  - gPROMS (unit operation level); ✓
  - SuperPro Designer (process flow sheet level);
  - GAMS (supply chain level).

# **Introduction to SuperPro Designer**

- Process flow sheet level modelling;
- Facilitates modelling, evaluation and optimization;
- Graphical user interface;
- Works by solving material and energy balances;
- Can be interfaced with Visual Basic for Applications (VBA), MatLab, Python, for automation and sensitivity analysis.





# SuperPro Designer – Key features

- Contains models for over 140 unit procedures/operations;
- Extensive chemical component and mixture database;
- Extensive equipment and resource databases;
- Equipment sizing and costing;
- Thorough process economics;
- Scheduling of batch operations;
- Throughput Analysis and Debottlenecking.





# SuperPro Designer – Typical questions being addressed

- How would the cost per dose change with increasing titres/yields?
- Which process design is more cost effective (by comparing different process designs)?
- How would the cost per dose and the upfront capital cost change at different production scales?
- Where are the production bottlenecks?
- Which are the major cost components and how to reduce those?
- How feasible is a continuous process compared to a batch process?





### SuperPro Designer – Graphical User Interface (GUI)



## SuperPro Designer – Model inputs

- Type of unit procedures and parameters for each operation within procedures;
- Material inputs (default values are provided for labour and utilities);
- Definition of the reaction/fermentation (e.g. stoichiometric, kinetic, equilibrium) and reaction components;
- Specification of material flows between unit procedures;
- Specification of the sequence and duration of unit procedures (i.e. scheduling);
- Optional: a variety of other inputs can be specified (e.g. costs of equipment, costs of consumables, costs of labour and utilities, etc.)

# SuperPro Designer – Outputs

Charts:

- Equipment occupancy charts
- Operations Gantt Charts
- Equipment Gantt Charts
- Material, labour and utility utilization charts

Reports (MS Word, MS Excel or PDF formats):

- Materials & Streams
- Economic Evaluation
- Cash Flow Analysis
- Itemized Cost
- Environmental Impact, etc.



# SuperPro Designer Example Problem 1 "Cost modelling for new vaccine processes"

### **RNA vaccines – how expensive? how to reduce costs?**



**Imperial College** 

London

- ~10 kb self-amplifying RNA
- produced by *in vitro* transcription
- cell-free product
- Co-transcriptional 5' capping, ARCA



### **RNA vaccines – how expensive? how to reduce costs?**

• ~10 kb self-amplifying RNA

**Imperial College** 

London

- produced by *in vitro* transcription
- Co-transcriptional 5' capping, ARCA
- 5x Recycling of materials


### **RNA production model in SuperPro Designer**



- ~10 kb self-amplifying RNA
- produced by *in vitro* transcription
- cell-free product

**Imperial College** 

London

• Co-transcriptional 5' capping, ARCA

Cost per dose at 20 µg/dose at 1 L reaction volume: 0.72 USD/dose \*

\* Costs do not include formulation and secondary manufacturing (fill and finish) costs, upstream costs were accounted for indirectly. Default capital, labour, consumables, utilities, facility-related, maintenance and QA/QC costing values and methods from SuperPro Designer were used.



#### **RNA Cost modelling results**



\* Costs do not include formulation and secondary manufacturing costs, upstream costs accounted indirectly; Simulation results with default costing values and method from SuperPro Designer, capital costs were spread over 5 years.

#### Imperial College London

### **RNA Cost modelling results**



\* Costs do not include formulation and secondary manufacturing costs, upstream costs accounted indirectly; Simulation results with default costing values and method from SuperPro Designer, capital costs were spread over 5 years.

### **RNA Cost modelling results**





#### **Getting Help**

• Pressing the F1 button;

• SuperPro Designer manual: <u>http://www.intelligen.com/downloads/SuperPro\_ManualForPrinting\_v10.pdf</u>

• Training videos, from the Intelligen, Inc. website: <u>http://www.intelligen.com/videos.html</u>

• Papers and literature, from the Intelligen, Inc. website: <u>http://www.intelligen.com/literature.html</u>

• Attending training course offered by Intelligen, Inc.: <u>http://www.intelligen.com/training.html</u>





- Work Stream 1 (WS1): Improving existing processes and platforms
- Work Stream 2 (WS2): Design, development and implementation of new platform technologies



- Modelling tools:
  - gPROMS (unit operation level); ✓
  - SuperPro Designer (process flow sheet level); ✓
  - GAMS (manufacturing and supply chain level).



#### Outline

Introduction to GAMS



- GAMS Example Problem
- Exercise

Latest GAMS version (demo) free download link:

http://www.gams.com/download/

Reference: "GAMS-A User's Guide"



### **Introduction to GAMS**

## <u>General Algebraic Modelling System</u>



Algebraic Modelling Language (AML)

(high-level computer programming language)



The **syntax** is **similar** to the **mathematical notation** of optimization problems.

The **programming** of the optimization model is **independent** of the **solution algorithm**.



### **How GAMS works**

GAMS is a modeling system for optimization that **provides an interface with a variety of optimization algorithms** (solvers).

Users introduce the model to GAMS in the form of algebraic equations.

GAMS compiles the model and interfaces automatically with a solver.



#### Imperial College London

#### **GAMS user interface**

#### gamside: \\icnas4.cc.ic.ac.uk\zk1210\gamsdir\projdir\gmsproj.gpr

– 0 ×

E C C C C C C C C C C C C C C C C C C C		× 18
Vicnas4.cc.ic.ac.uk\zk1210\gamsdir\projdir\Multi_obj_	veights.lst	No active process
coursework-M00-Gonzalo-Guillen-Gosalbez_v2.lst Multi_obj_w	eights.log   Multi_obi_weights.lst   Multi_obi_weights.gms   coursework-MOD-Gonzalo-Guillen-Gosalbez_v2.gms	coursework-moo-gonzalo-guillen-gosalbez_v2 multi_obj_weights
Compilation Equation Listing SOLVE example2 Using LP Fr	BAMS 25.1.3 r4e34d435fbd Released Oct 30, 2018 WEX-WEI x86 64bit/MS Windows 12/11/18 15:01:39 Page 1	Cplex Time: 0.00sec (det. 0.01 ticks)
E - Equation     Column Listing SOLVE example2 Using LP Fr     Column	Compilation	Optimal solution found. Objective : -18.300000
Model Statistics SOLVE example2 Using LP Fr     Solution Report SOLVE example2 Using LP Fr     SolEQU	1 * Solve the Example-1 problem with the most appropriate solver	Restarting execution Multi_obj_weights.gms(81) 2 Mb
B− SolVAR − Execution E− Display	2 3	Reading solution for model example2 Executing after solve: elapsed 0:00:03.750
Equation Listing SOLVE example2 Using LP Fr E- Equation	4 SETS 5 1 /1*4/	Mult_obj_weights.gms(/9) 3 Mb Generating LP model example2 Multi obj weights.gms(8) 3 Mb
Column Listing SOLVE example2 Using LP Fr Column Model Statistics SOLVE example2 Using LP Fr	6 j /1*2/ 7 k /1*11/;	LOOPS k = 11 7 rows 5 columns 16 non-zeroes
Solution Report SOLVE example2 Using LP Fr     SolEQU     SolVAR	o PARAMETERS	Executing CPLEX: elapsed 0:00:03.803
⊢ Execution ⊕- Display	10   101   11   1   -3	Cplex 12.8.0.0
Equation Listing SOLVE example2 Using LP Fr     Equation     Column Listing SOLVE example2 Using LP Fr	13   3   12   14   4   4   7	Reading data Starting Cplex
Column     Model Statistics SOLVE example2 Using LP Fr     Solution Report SOLVE example2 Using LP Fr	15 16 w1(k)	Space for names approximately 0.00 Mb Use option 'names no' to turn use of names off
€- SolEQU €- SolVAR	17 /1 0.01 18 2 0.1	CFXPARAM_Advance 2 CFXPARAM_Simplex_Display 2 CFXPARAM_Simplex_Limits Iterations 200000000
⊢ Execution     ⊡ Display     ⊢ Equation Listing SOLVE example2 Using LP Fr	19 3 0.2 20 4 0.3	CPXPARAM TimeLimit 1000 CPXPARAM Threads 1
Equation Column Listing SOLVE example2 Using LP Fr	21 5 0.4 22 6 0.5	CFXPARAM_Farallel 1 CFXPARAM_Tune_TimeLimit 200
Herein Column     Model Statistics SOLVE example2 Using LP Fr     Solution Report SOLVE example2 Using LP Fr	23 7 0.6 24 8 0.7	Tried aggregator 1 time. LP Presolve eliminated 3 rows and 3 columns. Reduced LP has 4 rows 2 columns and 2 populations
E SolEQU SolVAR	25 9 0.8 26 10 0.9	Presolve time = 0.00 sec. (0.00 ticks) Using devex.
E Display Equation Listing SOLVE example2 Using LP Fr	27 11 0.99 / 28	LP status(1): optimal Cplex Time: 0.00sec (det. 0.01 ticks)
Column Listing SOLVE example2 Using LP Fr	29 parameter w /0.01/; 30	Optimal solution found.
Model Statistics SOLVE example2 Using LP Fr     Solution Report SOLVE example2 Using LP Fr     SolEOLI	31 TABLE 32 a(i,j) 'Coefficients'	Restarting execution
Execution	33 1 2 34 1 -1 -1	Multi_obj_weights.gms(81) 2 Mb Reading solution for model example2
	35 2 -2 1 36 3 2 1	Executing after solve: elapsed 0:00:04.116 Multi_obj weights.gms(79) 3 Mb
Column Listing SOLVE example2 Using LP Fr Column Model Statistics SOLVE example2 Using LP Fr	37 4 3 -2; 38	Job Multi_obj_weights.gms Stop 12/11/18 15:01:43 elapsed 0:00:04.121
- Solution Report SOLVE example2 Using LP Fr - SolEQU	39 40 * Define variables xl , x2 and f	
E-SolVAR Execution	41 VARIABLES	Close Onen Loo C Summary only V Update
1.1	her and a second s	

### **GAMS optimization model types and solvers**

#### Main optimization model types in GAMS:

- LP Linear Programming
- NLP NonLinear Programming
- MIP Mixed Integer linear Programming
- **RMIP** relaxed MILP where the integer variables are treated as continuous
- MINLP Mixed Integer NonLinear Programming; involve integer variables and nonlinear equations
- **RMINLP** Relaxed MINLP where the integer variables are treated as continuous

#### Main solvers for each type of optimization model:

- LP CPLEX
  MIP, RMIP CPLEX
  NLP BARON, MINOS, CONOPT, ANTIGONE
- MINLP BARON, DICOPT, ANTIGONE



#### **Structure of a GAMS Model**

#### **SETS**

Declaration Assignment of members

#### PARAMETERS, TABLES, SCALARS (DATA)

Declaration Assignment of values

#### VARIABLES

Declaration Assignment of type Assignment of bounds and/or initial values (optional)

#### **EQUATIONS**

Declaration Definition

**MODEL and SOLVE statements** 

**OPTION** for output file and solver

**DISPLAY** statements (optional)



#### **Variables**

• Types of variables:

Variable Type	Allowed Range of Variables		
FREE	-∞ to +∞		
POSITIVE	0 to +∞		
NEGATIVE	-∞ to 0		
BINARY	0 or 1		
INTEGER	0, 1,, 100		

- The default type is **FREE**
- <u>Remark</u>: The variable being optimized must be FREE and not indexed



#### **Equations**

The keyword **EQUATIONS** is for **listing the names** (which are random) of the constraints and objective function.

The equations are **defined** by: equation\_name..

Syntax for the equality and inequality signs:

The basic arithmetic operators are:

+	addition	The somi colon :
-	subtraction	in peoded at the
*	multiplication	is needed at the
1	division	end of each
**	exponent	equation.
	ation nome veriable 1 + veria	blo $2 - E = 1$

**Example:** equation\_name.. variable\_1 + variable\_2 =E= 1;4

#### Imperial College London

### **Bounds and Initial Values**

Bounds and initial values can be provided by adding a **suffix** to the variables.

#### Syntax for specifying bounds and initial values:

(variable name).LO lower bound (e.g., x.LO)
(variable name).UP upper bound (e.g., x.UP)
(variable name).FX fixed value
(variable name).L level value, meaning actual value (initial or final)
(variable name).M dual prices, Lagrange or Kuhn-Tucker multipliers

No need to specify lower bounds of zero for variables defined as **POSITIVE VARIABLE**.

In general, it is not a requirement to specify initial values for the variables.

However, for **nonlinear** models it is often advisable to provide an **initial guess** (e.g.,  $X \cdot L=4$ ;).



### **Keywords**

#### MODEL

The keyword **MODEL** is used to **name the optimization model** and to **specify** which equations are included in it.

> OR MODEL (model name) /eq1,eq2/;

#### **OPTION**

The keyword **OPTION** is used to **suppress output** for debugging the compilation of the equations, and to **set options for solvers** (max CPU s).

#### SOLVE

The keyword **SOLVE calls the optimization solver**. The syntax is as follows:

SOLVE (model name) USING (solver type) MINIMIZING (objective variable);

#### DISPLAY

The keyword **DISPLAY** shows the values of the requested symbols.

DISPLAY (variable name).suffix; and DISPLAY (parameter or set name);



# GAMS Example Problem 1 Transportation problem





#### **Objective** $\rightarrow$ minimize transportation cost

Subject to: demand satisfaction & supply constraints

Indices (or sets):

i : plants (San Diego, Seattle)j : markets (New York, Topeka, Chicago)

Given Data (or parameters):

- $a_i$ : supply of commodity of plant *i* (in cases)
- $\boldsymbol{b}_i$ : demand for commodity at market *j* (cases)
- $c_{ij}$ : transportation unit cost between plant *i* and market *j* (\$/case)

Di				
Plants	New York	Chicago	Topeka	supply
Seattle	2.5	1.7	1.8	350
San Diego	2.5	1.8	1.4	600
demand	325	300	275	

Transportation costs  $\rightarrow$  \$90/case/kMile. ---

------



#### **Decision Variables:**

 $|\mathbf{X}_{ij}|$ : amount to transfer from plant *i* to market *j*.  $x_{ij} \ge 0$ ,  $\forall i, j$ 

#### Objective Function:

$$\text{Minimize} \sum_{i} \sum_{j} c_{ij} x_{ij} \text{ ($K)}$$

**Constraints:** 

Supply limit in plant *i*: 
$$\sum_{j} x_{ij} \le a_i, \forall i$$

Satisfy demand at market 
$$j: \sum_{i} x_{ij} \leq b_j, \quad \forall j$$

#### Imperial College London

#### **Representation in GAMS (Input)**

Sets i plants / seattle, san-diego / i markets / new-york, chicago, topeka / ; Parameters a(i) capacity of plant i in cases / seattle 350, san-diego 600 / b(j) demand at market j in cases / new-york 325 chicago 300 topeka 275 / ; Table d(i,j) distance in thousands of miles new-york chicago topeka seattle 2.5 1.7 1.8 1.8 1.4 ; san-diego 2.5 Representation freight in dollars per case per thousand miles /90/; Scalar f Parameter c(i,j) transport cost in thousands of dollars per case ; in GAMS c(i,j) = f\*d(i,j)/1000;Variables x(i,j) shipment quantities in cases (Input) total transportation costs in thousands of dollars ; Z Positive Variable x ; z.lo = 1; Equations define objective function cost supply(i) observe supply limit at plant i demand(j) satisfy demand at market j ; z = e = sum((i,j), c(i,j)\*x(i,j));cost .. supply(i) .. sum(j, x(i,j)) = l = a(i);demand(j) .. sum(i, x(i, j)) = g = b(j);Model transport /all/ ; Solve transport Using LP Minimizing z; Display x.L, x.M;



#### **General notes**

- Declare sets and variables first! You cannot refer to something that has not been defined!
- Terminate statements with semi-colons (;)
- GAMS compiler is not case-sensitive
- Lines starting with \* are comment lines
- Names must start with a letter
- Descriptive text must:
  - fit on one line, and be no more than 80 characters long
  - not start with GAMS' reserved words or contain the symbols =, ;/



#### **Sets**

#### • Sets in GAMS ≡ indices in algebraic models, e.g.,

Sets	i	canning plants	/	seattle, s	san-diego	/		
	j	markets	/	new-york,	chicago,	topeka	/	;

- **Multi-words not allowed**: NEW-YORK not NEW YORK
- Can also write as:

Set	i	canning plants	/	seattle,	san-diego	/		
Set	j	markets	/	new-york,	chicago,	topeka	/	;

• Use of the asterisk in set assignment:

 Set
 T time periods
 / 1991\*2000 /

 M machines
 / MACH1\*MACH24 /;

This corresponds to T = {1991, 1992, ..., 2000} and M = {MACH1, MACH2, ..., MACH24}

ALIAS (I, IP); defines the SET (index) IP identical to the SET I



#### **Input Data**

• Entry by lists:

Parameters a(i) capacity of plant i in cases
/ seattle 350
san-diego 600 /
b(j) demand at market j in cases
/ new-york 325
chicago 300
topeka 275 / ;

• Entry by Tables:

<b>Table</b> d(i,j)	distance	in thousands	of miles		
		new-york	chicago	topeka	
seatt	le	2.5	1.7	1.8	
san-d	iego	2.5	1.8	1.4	;
Alternatively: d ( ``	seattle",	"new-york")=2	.5; etc.		

• Entry by Direct Assignment:

```
Parameter c(i,j) transport cost in thousands of dollars per case ;
    c(i,j)=f*d(i,j)/1000;
```

• Zero is default value for all un-assigned parameters and scalars



#### **Equations**

- Note: equations must be declared and then defined
- Use of SUM and PROD operators:



Also, they can be used in direct assignment of PARAMETERS and SCALARS, e.g.,

SCALAR TOTSUPPLY total supply over all plants; TOTSUPPLY = SUM(i, b(i));



#### The Dollar Operator (\$)

**Provides a concise exception-handling capability** 

- Example 1: If y ≥ 1.5, then x = 2, else x = 1 SCALAR X,Y; Y = 2; X = 1; x = 2\$ (Y GE 1.5);
- Example 2: If y ≤ 1.5, then x = 2, else x = 1 SCALAR X, Y; Y = 2; X = 1; x\$ (Y LE 1.5) = 2;
- **Example 3:** If  $x_i \neq 0$ , then  $\rho_i = 1/x_i$ , else  $\rho_i = 0$ rho(i) = (1/x(i))\$(x(i) NE 0);
- **Example 4** (Equations):  $j = \{1, 2, 3, 4\}$ .  $\sum_{j=2}^{4} x_j = 1$ EQ1.. SUM(j\$(j>1), x(j)) = E = 1; j=2
- **Example 5** (Equations): If  $x_i \neq 0$ , then  $z_i = y_i-3$ EQ2\$(x(i) NE 0).. z(i) =E= y(i) - 3;



### **GAMS Compilation**

- Open GAMS IDE
- Create a **GAMS project file (project name).gpr**
- Create a **GAMS input file (file name).gms** and save it in the same directory with the project file
- Run GAMS by pressing F9 or the run button in the GAMS IDE
- After the compilation of the .gms file, an output file is created (file name).lst



### **The GAMS Output**

### The main elements of the GAMS output are:

- Echo Print
- Error Messages
- Equation Listings
- Model Statistics
- Status Reports
- Solution Reports



### **Solution Reports**

• The input statement **DISPLAY x.L, x.M**;



san-diego 325.000 275.000

	65	VARIABLE x.M	shipment	quantities	in cases
		new-york	chicago	topeka	
seattle		EPS		0.03	86
san-diego	C		0.009	9	



#### **Final Remarks**

Some other useful operators are the following:

- Dollar operator (\$): it can be used to restrict the elements of a set; roughly speaking, it has the function of command IF.
- LOOP keyword which can be used to perform repetitive calculations for parameters or sets.
- FOR can be used for multiple SOLVE statements in an iterative algorithm.
- There are also other usual programming commands; e.g., **IF** and **WHILE**.

GAMS is available for several operating systems (Windows, Linux, Mac, Solaris). GAMS is free, but the demo version could solve problems of limited size. Solvers are NOT free (license is needed for every solver).

Other Algebraic Modeling Languages: AMPL, AIMMS, etc.

#### Imperial College London

### **Example Problem 2**

#### A maximum-profit production problem

A furniture company wants to maximize its profits from the manufacture of four different types of desks.

Each different desk type requires different number of man-hours in each of the company sections, carpentry and finishing.

The profit per unit of every different type of desk sold is different and given.

The capacity of man-hours for carpentry and finishing is also given.

#### Labour requirements (man-hours)

	d1	d2	d3	d4
Carpentry	4	9	7	10
Finishing	1	1	3	40

Profit per unit sold					
d1	d2	d3	d4		
12	20	18	40		

Capacity (m	nan-hours)
Carpentry	6000
Finishing	4000

Formulate the optimization problem, and then introduce it to GAMS and solve it.



# **Problem Formulation**

Sets: *i*: desk type (d1, d2, d3, d4) *j*: sector (carpentry, finishing)

Data: caplim (*j*) : capacity limit of sector *j* profit (*i*) : profit form selling desk type *i* labor (*j*, *<u>i</u>) : labor requirements (man-hours)* 

Decision Variable:  $x_i$ : number of desks *i* produced

Objective Function :  $\sum_i x_i profit_i$ 

Constrain:  $\sum_{i} labor_{j,i} x_i \leq caplim_j \forall j$ 

#### **Representation in GAMS (Input)**

```
Sets desk / d1, d2, d3, d4 /
     shop / carpentry, finishing /
Table labor(shop,desk) labor requirements (man-hours)
           d1 d2 d3 d4
 carpentry 4 9 7 10
 finishing 1 1 3 40
Parameters
  caplim(shop) capacity (man hours) / carpentry = 6000, finishing = 4000 /
  price(desk) per unit sold ($) / d1 = 12, d2 = 20, d3 = 18, d4 = 40 /
Variables
  mix(desk) mix of desks produced (number of desks)
                                     ($)
  profit total profit
Positive Variable mix
Equations
  cap(shop) capacity constraint (man-hours)
                                    ($);
  ap accounting: total profit
cap(shop).. sum(desk, labor(shop,desk)*mix(desk)) =l= caplim(shop);
ap.. profit =e= sum(desk, price(desk)*mix(desk));
Model pmp product mix problem / all /; Solve pmp maximizing profit using lp;
```



#### **Getting Help**

- GAMS User Guide
- Expanded GAMS Guide (McCarl)
- Example from Model Library

SeqNr	Name	Application Area	Туре	Contributor	Description	*	
001	TRNSPORT	Management Science and OR	LP	Dantzig, G B	A Transportation Problem		> Example
)02	BLEND	Management Science and OR	LP	Dantzig, G B	Blending Problem I		Example
103	PRODMIX	Management Science and OR	LP	Dantzig, G B	A Production Mix Problem		Example
104	WHOUSE	Management Science and OR	LP	Dantzig, G B	Simple Warehouse Problem		
105	JOBT	Management Science and OR	LP	Dantzig, G B	On-the-Job Training		
006	SROUTE	Management Science and OR	LP	Dantzig, G B	The Shortest Route Problem		
07	DIET	Micro Economics	LP	Dantzig, G B	Stigler's Nutrition Model		
108	AIRCRAFT	Management Science and OR	LP	Dantzig, G B	Aircraft Allocation Under Uncertain Demand		
109	PRODSCH	Management Science and OR	MIP	CDC	APEX - Production Scheduling Model		
10	PDI	Management Science and OR	LP	ARCNET	ARCNET - Production Distribution and Inventory		
11	UIMP	Management Science and OR	LP	Ellison, E F	UIMP - Production Scheduling Problem		
12	MAGIC	Management Science and OR	MIP	Garver, L L	Magic Power Scheduling Problem		
13	FERTS	Micro Economics	LP	Choksi, A M	Egypt - Static Fertilizer Model		
14	FERTD	Micro Economics	MIP	Choksi, A M	Egypt - Dynamic Fertilizer Model	-	
•		1					



### **Consider this problem**




### **Answer: 26 hours**





## Will a bioprocess design work?

Interconnected process steps

Complex purification

Many equipment items



#### Imperial College London **Can this site make enough of these new pharmaceuticals?**

Horizon		2_year_pharmaceuticals
01-Jan-01 00:00 -	(Ceiling/)	Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
- 30-Dec-02 00:00	Capacity	01 29 26 23 21 18 16 13 10 08 05 03 31 28 25 22 20 17 15 12 09 07 04 02
Bay_1	100	
Bay_2	100	
Bay_3	100	
Bay_4	100	
Bay_5	100	
		between products
A_1	10000/	res, but mese resources
A_2a	0.01/	are very busy!
A_2b	10000/	
A_3	10000/	
A_4	2000/	
A_5_6	5000/	
B_7	10000/	
B_8	5000/ 50000	
C_1	5000/ 50000	
C_2	2000/ 50000	
C_3	2000/ 50000	
C_4	10000/	

## **Conclusions**

- We are working together with developing country manufactures to improve their vaccine production processes;
- We are designing and implementing new vaccine production technologies;
- We are doing multi-level modelling using the following tools:
  - gPROMS (unit operation level);
  - SuperPro Designer (process flow sheet level);
  - GAMS (supply chain level).

# Imperial College London

Zoltan Kis, Maria Papathanasiou, Cleo Kontoravdi and Nilay Shah

