Engineering Notes

Computational Workflow Management for Conceptual Design of Complex Systems

L. K. Balachandran* and M. D. Guenov†
Cranfield University, Cranfield, MK43 0AL UK

DOI: 10.2514/1.43473

Nomenclature

c	=	column number in the incm
DSM	=	design structure matrix
GA	=	genetic algorithm
IMM	=	incidence matrix method
incm	=	incidence matrix
incmf	=	foundation incidence matrix
incmprod	=	product of nonzero elements of column c of incmf
MDO	=	multidisciplinary design optimization
m	=	number of columns of the incm
NIvar	=	number of independent variables
Nmod	=	number of models
Noutmod	=	sum of the total number of outputs of each model in a system
n	=	number of rows of the incm
nFdb	=	number of feedback loops
nMm	=	number of modified models
r	=	row number in the incm
SCC	=	strongly connected component
TNvar	=	total number of variables
valc(c)	=	product of nonzero elements of column c of incm
valcf(c)	=	product of nonzero elements of column c of incmf matrix, if incmprod is not equal to two
valr(r)	=	product of nonzero elements of row r of incm
valrf(r)	=	product of nonzero elements of row r of incmf

I. Introduction

OUR initial research, as part of an industry-led European Union aeronautical project, indicated that the early design stage would benefit, among other things, from a new approach that supports the exchange of mathematical models and simulation data, thus enabling an overall design optimization process with robust, flexible, and dynamic workflows. During this phase hundreds of low-fidelity models, together with thousands of variables, are used to describe a complex product such as aircraft. These models need to be rapidly assembled in an appropriate computational workflow in order to perform various design studies, including trade-off analysis and optimization. Currently, the derivation of the workflows in industry is carried out predominantly manually and is very time consuming. Furthermore, the process requires a great deal of additional skills and knowledge that are not directly related to the designing of the aircraft itself.

In this context, proposed is a novel automated approach whose main feature is the dynamic derivation of an effective and efficient computational workflow, given the designer's choice of input variables.

In a wider context, this work is related to the better known multidisciplinary design optimization (MDO) problem, i.e., solving and optimization of a system consisting of coupled disciplines. We have linked this work to state-of-the-art MDO at early design stage in [1]. Below we focus on the innovative features of the method.

II. Computational Process Modeling

Computational process modeling is the process of organizing a complex system of models in order to compute the output variables given the (selected by the designer) set of independent (input) variables.

In summary the algorithm works as follows. The first step, once the designers have chosen the independent variables, is to establish that the system is well constrained, i.e., that it is solvable and the inputs and outputs of all models are determined and consistent. Additionally, all models that due to the choice of independent variables ought to have inputs and outputs reversed are identified. The next step is to establish the existence of any strongly connected components (SCCs) resulting from models coupling due to shared variables. If SCCs exist, then these are subjected to the same consistency and determinacy procedures as described above. Furthermore, the internal sequence of the models in each SCC is rearranged with respect to minimizing the iterations when solving. Finally the SCCs are reintroduced into the overall computational system as if they were single models. The aggregated system is then further rearranged to obtain a workflow with minimum computational cost. Numerical treatments such as fixed-point iteration and the Gauss–Newton method are applied to the reversed models and the SCCs, respectively, upon solving. An object-oriented approach is applied [2] to enable an arbitrary level of processes nesting depending on the complexity of the design study. In the following subsections we briefly summarize the main components of the method.

A. Variable flow modeling

A novel incidence matrix method (IMM) is proposed here, which dynamically obtains the information flow within the system, i.e., its consistency.

An incidence matrix (incm) has models in the rows and variables in the columns. The algorithm for populating the incidence matrix is shown in Fig. 1.

At the start there are only ones and zeros as entries in the incidence matrix, representing the presence or absence of a variable in a model. Following the steps of the IMM (Fig. 1), the ones are replaced with either twos (represents an input) or threes (represents an output) thus obtaining the variable flow model of the system. The parameters used in the flow chart are given in Eqs. (1–4):

$$\mathrm{valr}\,2(r,c)=\frac{\log(\mathrm{valrf}(r)/\mathrm{valr}(r))}{\log(2)} \tag{1}$$

$$\mathrm{valr}\,3(r,c)=\frac{\log(\mathrm{valrf}(r)/\mathrm{valr}(r))}{\log(3)} \tag{2}$$

$$\mathrm{valc}\,2(r,c)=\frac{\log(\mathrm{valcf}(c)/\mathrm{valc}(c))}{\log(2)} \tag{3}$$

Received 28 Jan. 2009; revision received 2 Oct. 2009; accepted for publication 12 Dec. 2009.

*Research Fellow, Advanced Engineering Design Group, Department of Aerospace Engineering.

†Professor, Advanced Engineering Design Group, Department of Aerospace Engineering. Member AIAA.

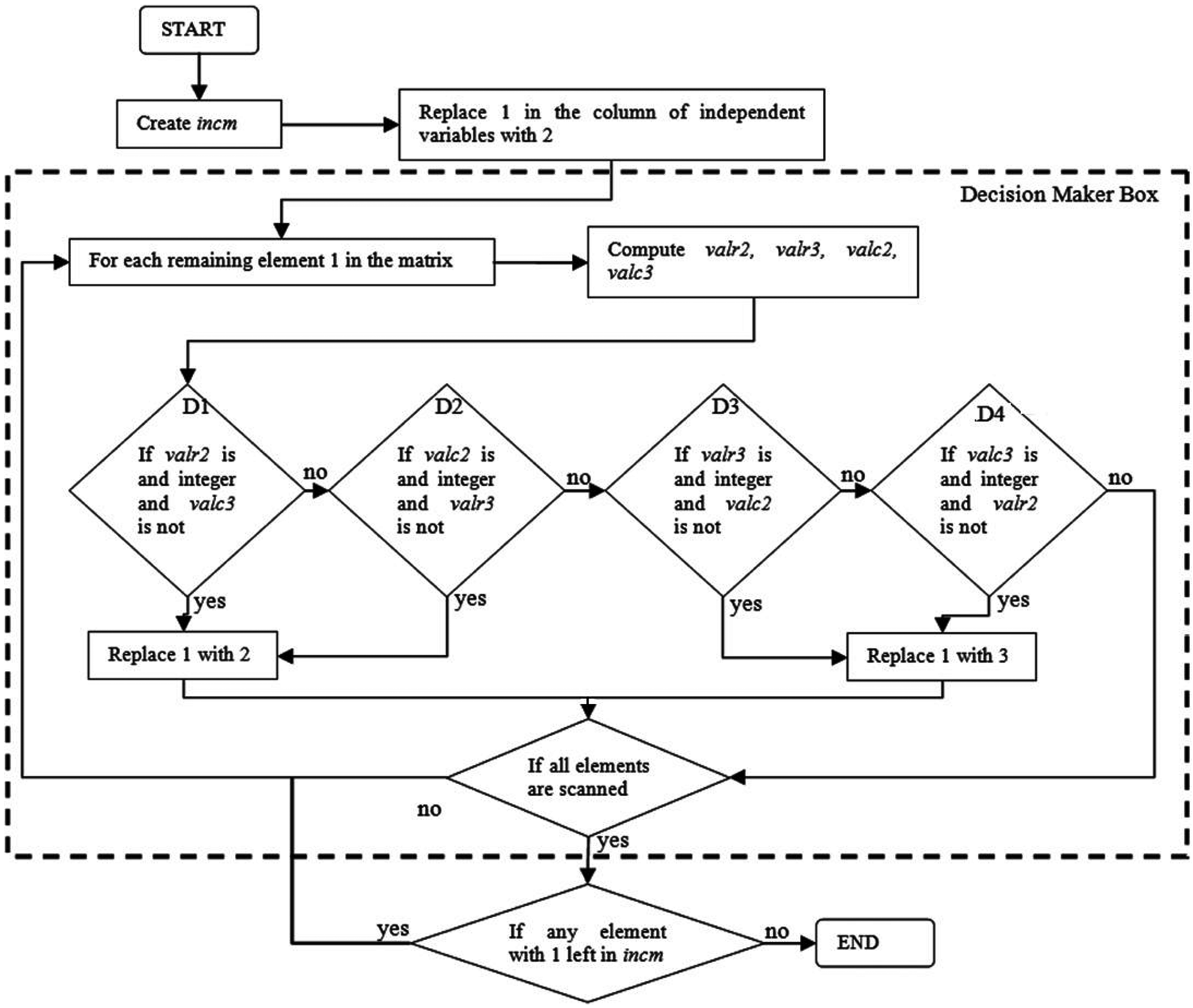

Fig. 1 Flow chart for the incidence matrix method.

$$\text{valc}\,3(r,c) = \frac{\log(\text{valcf}(c)/\text{valc}(c))}{\log(3)} \tag{4}$$

The unknown variables in the right-hand side of Eqs. (1–4) are given in Eqs. (5–8):

$$\text{valrf}\,(r) = \prod_{c=1}^{m} \text{incmf}(r,c); \qquad \text{incmf}(r,c) \neq 0 \tag{5}$$

$$\text{valcf}(c) = \begin{cases} 3; & \text{incmprod} = 2 \\ \text{incmprod}; & \text{incmpro} \neq 2 \end{cases}$$

where,

$$\text{incmprod} = \prod_{r=1}^{n} \text{incmf}(r,c); \qquad \text{incmf}(r,c) \neq 0 \tag{6}$$

$$\text{valr}\,(r) = \prod_{c=1}^{m} \text{incm}(r,c); \qquad \text{incm}(r,c) \neq 0 \tag{7}$$

$$\text{valc}\,(c) = \prod_{r=1}^{n} \text{incm}(r,c); \qquad \text{incm}(r,c) \neq 0 \tag{8}$$

In Eqs. (1–8) incmf contains the original inputs and outputs of the model. The entries of incmf are either two or three, marking the inputs and outputs of the original model, respectively, or zero for no relation.

B. Variable flow modeling in the presence of SCCs

Generally, situations may arise in which the independent variables specified by the designer are either too many or too few for the system to produce a fully populated incm after applying the IMM. In the former case the system is overdetermined while in the latter case the system is underdetermined. Both overdetermined and underdetermined systems lead to a partially populated incm after applying the IMM; i.e., some of the ones still remain in the matrix. Underdetermined systems are resolved by defining additional independent variables. Overdetermined systems can be resolved by removing variables from the group of independent variables. (This, of course, cannot be done arbitrarily since the design space is affected.)

There are cases in which incm remains partially populated even though the system is determined. This is due to the presence of SCCs. Such partially populated incm is resolved by guessing the inputs and output variables of any one of the models belonging to the SCC according to the following rule:

Among the models which are part of a SCC for which not all ones have been replaced so far, the models for which the new inputs differ from the original ones are selected for guessing. If no such model exists, the incm is populated with the original inputs and outputs of the models.

This rule limits the unnecessary generation of modified models and alternative variable flow models.

Table 1 Results for Case 1

Computational process modeling				Solving	
Variable flow model	nFdb	nMm	Optimal flow model as chosen by our method	Number of calls to the models in SCC	% additional computational cost
1	3	6		117	95% more
2	5	11		158	163% more
3	6	3	•	60	Base
4	5	9		198	230% more
5	8	11		Nonconverged	-

Once the inputs and outputs of the model are guessed based on the above rule, the incm is further iterated through the IMM to obtain the fully populated matrix.

C. Choosing the independent variables

Unlike a system of algebraic equations, the models (black boxes) considered in this work can have multiple outputs, and therefore a modified criteria for solvability of the system is required. We propose such criteria in Eq. (9):

$$\begin{aligned}&\text{TNvar-NIvar-Noutmod} = 0 \rightarrow \text{determined system}\\&\text{TNvar-NIvar-Noutmod} > 0 \rightarrow \text{underdetermined system}\\&\text{TNvar-NIvar-Noutmod} < 0 \rightarrow \text{overdetermined system}\end{aligned} \tag{9}$$

The equation can be rewritten as

$$\text{Nmod} = \text{TNvar-NIvar-(Noutmod-Nmod)} \tag{10}$$

The termNoutmod-Nmod in Eq. (10) takes into account the multiple outputs generated by certain models. Since the multiple outputs can be calculated simultaneously once that model's inputs are known, these outputs are considered as a single unknown variable. Thus, if Eq. (10) is satisfied then the system is determined.

D. System decomposition

This is the process of decomposing a complex system into a number of subproblems, including the identification of SCCs. We have adapted an algorithm by Tang et al. [3] to deal with this problem.

E. Scheduling the SCCs

The purpose of scheduling is to sequence the models so that these can be executed as a computational workflow. Presence of feedback loops in the SCCs makes it necessary to employ iterative methods for their solution. Therefore, reducing the number of feedback loops (and their length) will reduce the computational cost.

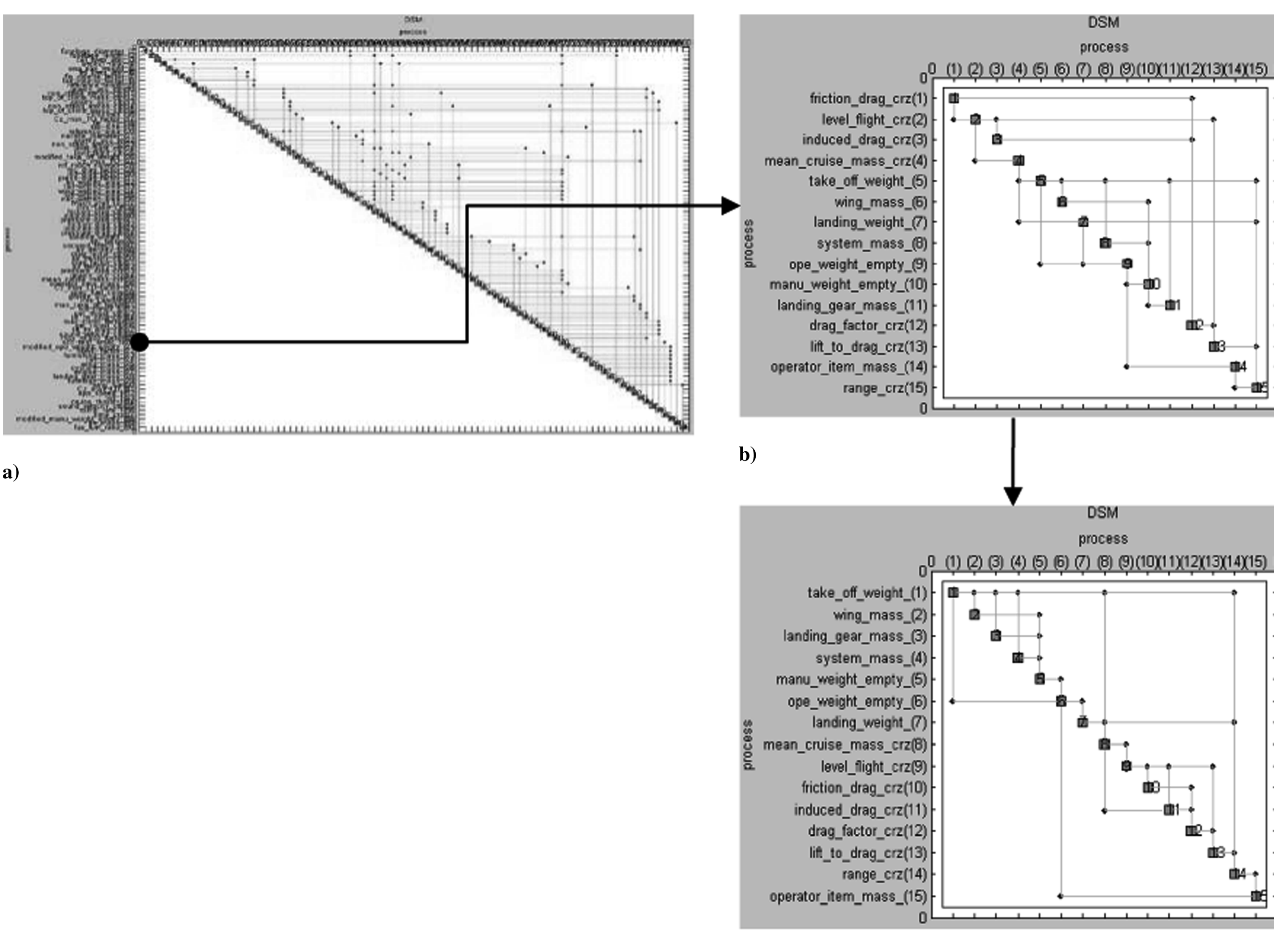

Fig. 2 a) DSM of Case 1, b) DSM of the SCC of Case 1, and c) optimized DSM of the SCC of Case 1.

Table 2 Results for Case 2

Computational process modelling					Solving	
SCC	Variable flow model	nFdb	nMm	Optimal flow model (chosen by our method)	Number of calls to the models in SCC	% additional computational cost
SCC 1	1	1	4	•	110	Base
	2	1	5		110	Equal
SCC 2	1	1	3		74	Base
	2	1	4		320	332% more
	3	1	2	•	86	16.2% more

Our proposed sequencing method is based on an approach [4] that uses a genetic algorithm for ordering of complex design processes. The number of feedback loops (nFdb) is chosen as the objective function to be minimized as shown in Eq. (11), where D is the DSM populated from the incidence matrix of a SCC and n is the size of the matrix:

$$\mathrm{nFdb} = \sum_{i=2}^{n} \sum_{j=1}^{i-1} D(i, j) \tag{11}$$

F. Scheduling the entire system

The models which belong to a SCC, which have already been arranged using the method described in the previous subsection, are now considered as single models and are arranged together with the remaining noncoupled models. Applying again the algorithm of Tang et al. [3], the global DSM is rearranged into an upper-triangular matrix, thus ensuring that all loops are feedforward.

III. Results

The testing of the method has been conducted on a simplified aircraft model supplied by our industrial partners. It contains 97 models and 125 variables. Despite being significantly simplified, the test case still produces an aircraft description that can be judged by experts [1].

We ran more than 100 test cases with random combinations of 23 input variables to ensure that the system is determined. In all tests the SCCs were solved by fixed-point iteration while the modified models were resolved by Gauss–Newton method [5]. Discussed here for brevity are only two representative cases.

A. Case 1

After decomposition, 13 out of the 97 models formed a SCC for which 12 variable flow models were generated. Only four of these produced a converged solution. On the whole, the nonconverged variable flow models contained higher values for number of modified models (nMm) and nFdb. Table 1 provides details for 5 of the 12 variable flow models, including one that did not converge and is shown only for comparison.

From Table 1 it is seen that when nMm increases, number of calls to the models in the SCC (i.e., the computational cost) also increases. However this does not hold for variable flow model 4. This discrepancy can be attributed to the fact that the convergence of the SCC was not only depending on nMm and nFdb but also on other factors, such as the starting iteration point for the unknown variables, mutual sensitivity of the reversed input and output variables of the modified models, and possible other factors yet to be discovered. (Although convergence is an important issue it falls outside the scope of this publication).

For illustration, shown in Fig. 2a is a DSM that corresponds to the first variable flow model in Table 1. Black dots above the diagonal denote feedforward loops, and dots below the diagonal represent feedback loops.

B. Case 2

In this case, two SCCs were identified (Table 2).

The sixth column (number of calls to the models in SCC) of Table 2 indicates that the optimum flow model selected for SCC 2, although close, was actually not the optimal one in terms of computational cost. Nevertheless, these and many more extensive tests demonstrated that the selections made by the proposed method were always among the best [6].

C. Current limitations

Model reversibility is a generic problem that we have assumed to be solvable with iteration/optimization methods. However, in some cases (e.g., periodic functions) there can be more than one output per given input. Despite that in general the low-fidelity models used for early aircraft design are well-behaved, the reversibility test must be carried out for each model (black box) before certification for dynamic computational design studies.

Another practical problem, which we consider manageable but have not addressed here, is the possibility of using more complex input/output data types such as matrices.

IV. Conclusions

Presented is a computational process modeler for dynamic composition and solving of systems of nonlinear models, considered as black boxes. The main purpose of this approach has been to automate a largely manual process currently followed by industry for early design, which requires a great deal of additional skills and knowledge that are not directly related to the designing of the aircraft itself.

As part of this framework, proposed is a method for ensuring that the system is well-constrained and which minimizes the computational effort upon solving. Other novel features include handling of models with multiple outputs as well as identification and computational optimization of subsets of models coupled through shared variables.

Extensive tests performed so far on a simplified but still representative industrial test case confirmed at large the effectiveness of the computational process modeler.

Future work will concentrate on issues related to the convergence of the SCCs and handling of compound data.

Acknowledgements

This research has been carried out within the *Value Improvement Through a Virtual Aeronautical Collaborative Enterprise* Integrated Project (AIP CT-2003-502917), which is partly sponsored by the Sixth Framework Program of the European Community under priority 4, Aeronautics and Space.

We would like to thank our industrial partners and, in particular, Thierry Druot and Yannick Caillabet from Airbus.

References

[1] Guenov, M. D., Fantini, P. F., Balachandran, L. K., Maginot, J. P., and Padulo, M., "MDO at Pre Design Stage," *Advances in Collaborative Civil Aeronautical Multidisciplinary Design Optimization*, edited by E. Kesseler and M. D. Guenov, AIAA Progress Series, AIAA, New York (to be published).

[2] Balachandran, L. K., and Guenov, M. D., "Object Oriented Framework for Computational Workflows in Air-Vehicle Conceptual Design," *9th AIAA Aviation Technology, Integration, and Operations Conference*, South Carolina, AIAA Paper 2009-7117, 21–23 Sept. 2009.

[3] Tang, D., Zheng, L., and Zhizhong, L., "Re-Engineering of the Design Process for Concurrent Engineering," *Computers and Industrial Engineering*, Vol. 38, No. 4, 2000, pp. 479–491. doi:10.1016/S0360-8352(00)00059-0

[4] Rogers, J. L., "Reducing Design Cycle Time and Cost Through Process Resequencing," *Proceedings of the 11th International Conference on Engineering Design*, Vol. 1, Tampere, Tampere, Finland, 1997, pp. 193–198.

[5] Dennis, J. E., and Schnabel, R. B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall, Upper Saddle River, NJ, 1983, pp. 221–228. doi:10.1137/1028131

[6] Balachandran, L. K., "Computational Workflow Management for Conceptual Design of Complex Systems: An Air-Vehicle Design Perspective," Ph.D. Dissertation, Department of Aerospace Engineering, Cranfield University, Cranfield, Bedfordshire, England, 2007.